

Guide Azure DevOps Services

Plan smarter, collaborate better,
and ship faster with a set of
modern dev services.



Contents

Azure DevOps user guide

Overview

- What is Azure DevOps?

- Overview of services

- Where's VSTS?

Quickstarts for users

- Code with Git

- Set up continuous integration & delivery

- Plan & track work

- Add & run manual tests

- View permissions

Quickstarts for admins

- Sign up for Azure DevOps

- Create organizations

- Add users to a project or team

- Configure team resources

- Get started as an admin

- Install a Marketplace extension

Tutorials for users

- Set favorites

- Follow work & pull requests

- Get started as a Stakeholder

Tutorials for admins

- Change individual permissions

- Grant or restrict permissions to select tasks

Concepts

- Key concepts

- Define organizations and projects

- Source control

[Backlogs, boards, & Agile tools](#)

[Clients & tools](#)

[Software development roles](#)

[What's the difference? Azure DevOps Services vs. TFS](#)

[Glossary](#)

[How-to guides](#)

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

[Troubleshooting](#)

[Troubleshoot connection](#)

[TF31002: Unable to connect](#)

[Troubleshoot network connections and whitelist addresses](#)

[Get support, provide feedback](#)

[Reference](#)

[Permissions & access \(Security\)](#)

[About access levels](#)

[Keyboard shortcuts](#)

[Navigate in Team Explorer](#)

[FAQs](#)

[Resources](#)

[Web portal navigation](#)

[Manage projects](#)

[Security & identity](#)

[Billing](#)

[Visual Studio IDE](#)

[Visual Studio Code](#)

[Visual Studio for Mac](#)

What is Azure DevOps Services?

9/10/2018 • 2 minutes to read • [Edit Online](#)

Azure DevOps Services

Azure DevOps Services is a cloud service for collaborating on code development. It provides an integrated set of features that you access through your web browser or IDE client, including the following:

- Git repositories for source control of your code
- Build and release management to support continuous integration and delivery of your apps
- Agile tools to support planning and tracking your work, code defects, and issues using Kanban and Scrum methods
- A variety of tools to test your apps, including manual/exploratory testing, load testing, and continuous testing
- Highly customizable dashboards for sharing progress and trends
- Built-in wiki for sharing information with your team

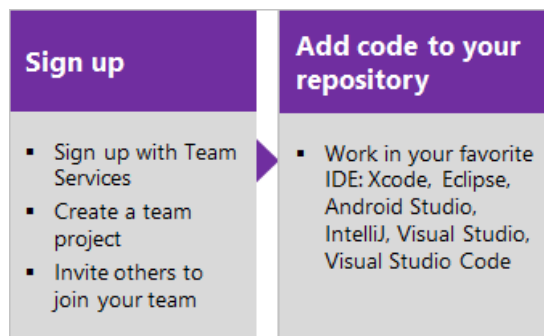
In addition, the Azure DevOps ecosystem provides support for adding extensions, integrating with other popular services, such as: Campfire, Slack, Trello, UserVoice, and more, and developing your own custom extensions.

Choose Azure DevOps Services when you want quick setup, maintenance-free operations, easy collaboration across domains, elastic scale, and rock-solid security. You'll also have access to cloud load testing, cloud build servers, and application insights.

Where do I start?

We've made it easy for you to start for free and try out our services.

Sign up for free by creating an organization and then, either upload your code to share or source control, or begin tracking your work using Scrum, Kanban, or a combination of methods.



Try this next

[Sign up and invite teammates](#)

Related articles

- [A tour of services](#)
- [Key concepts](#)
- [Client-server tools](#)
- [Software development roles](#)
- [Pricing - Azure DevOps](#)

What do I get with Azure DevOps Services or TFS?

9/11/2018 • 6 minutes to read • [Edit Online](#)

Azure DevOps Services | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

With Azure DevOps Services and TFS, you gain an integrated set of services and tools to manage your software projects, from planning and development through testing and deployment. Services are delivered through a client-server model, many of which are delivered through an easy-to-use web interface that 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.

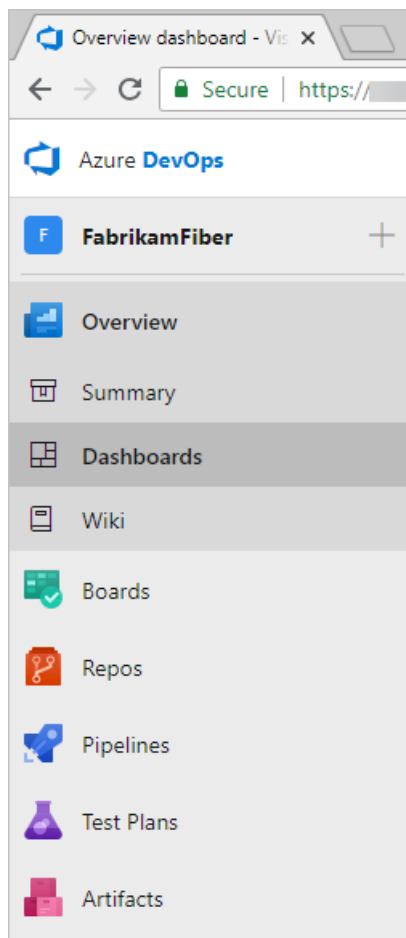
Access Web services through the following areas, as shown in the following image.

NOTE

The **New navigation** feature provides a vertical navigation experience and is in preview for Azure DevOps Services. When you [enable new navigation](#), you automatically enable several new Agile tool features that are described in the [New Work Hubs](#) blog post.

On-premises Microsoft Team Foundation Server users can select **Previous navigation** for guidance.

- [New navigation](#)
- [Previous navigation](#)



Many of our services are either free for small teams or available through a subscription model or per use model. Where needed, you can exercise a hybrid approach where you use an on-premises TFS to manage your code and

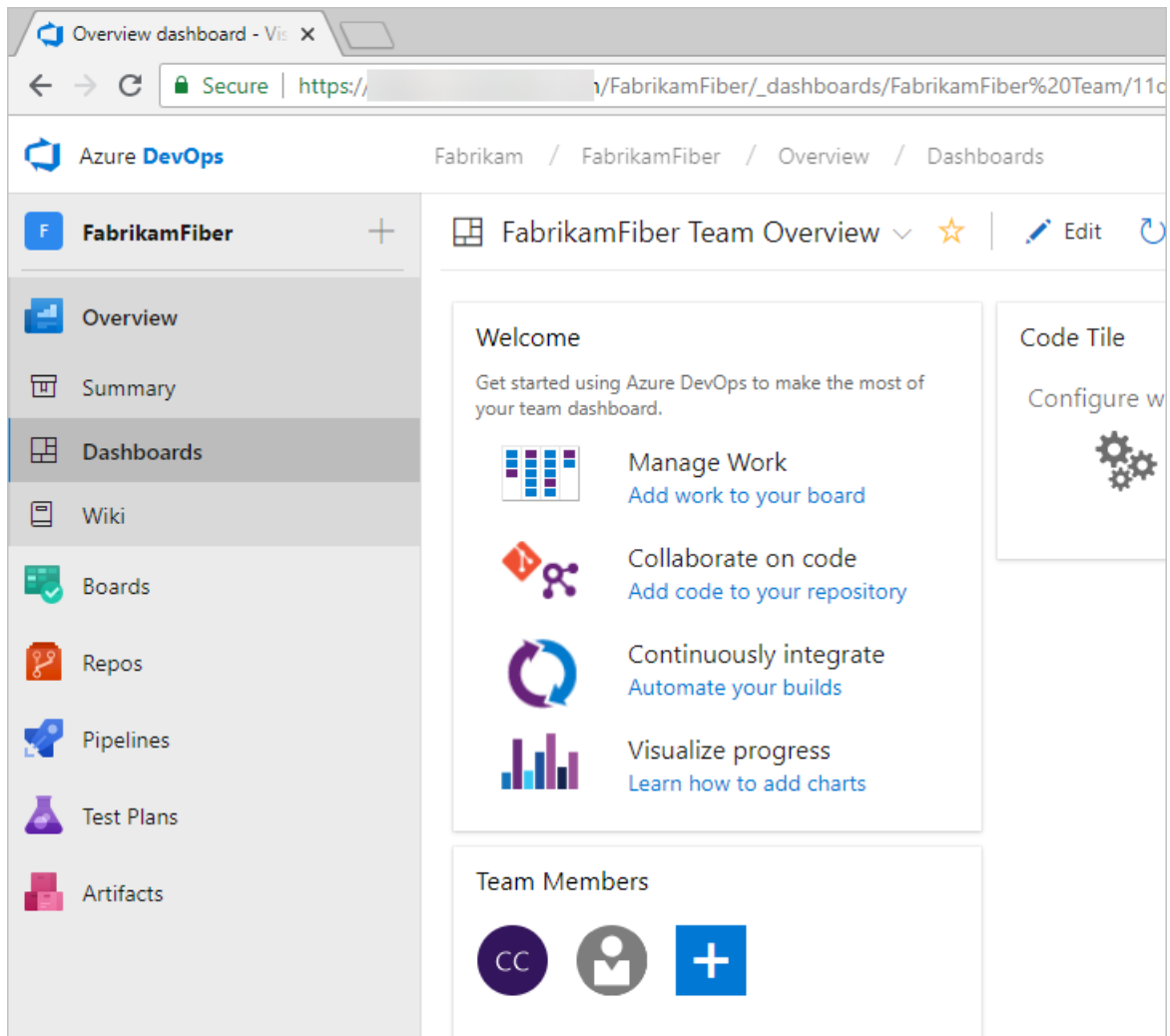
work, and purchase cloud build or testing services on an as-needed basis.

For information about client tools, see [Tools](#).

Dashboards

From **Dashboards** you gain access to user-configurable dashboards.

- [New navigation](#)
- [Previous navigation](#)



You can perform the following tasks in Dashboards:

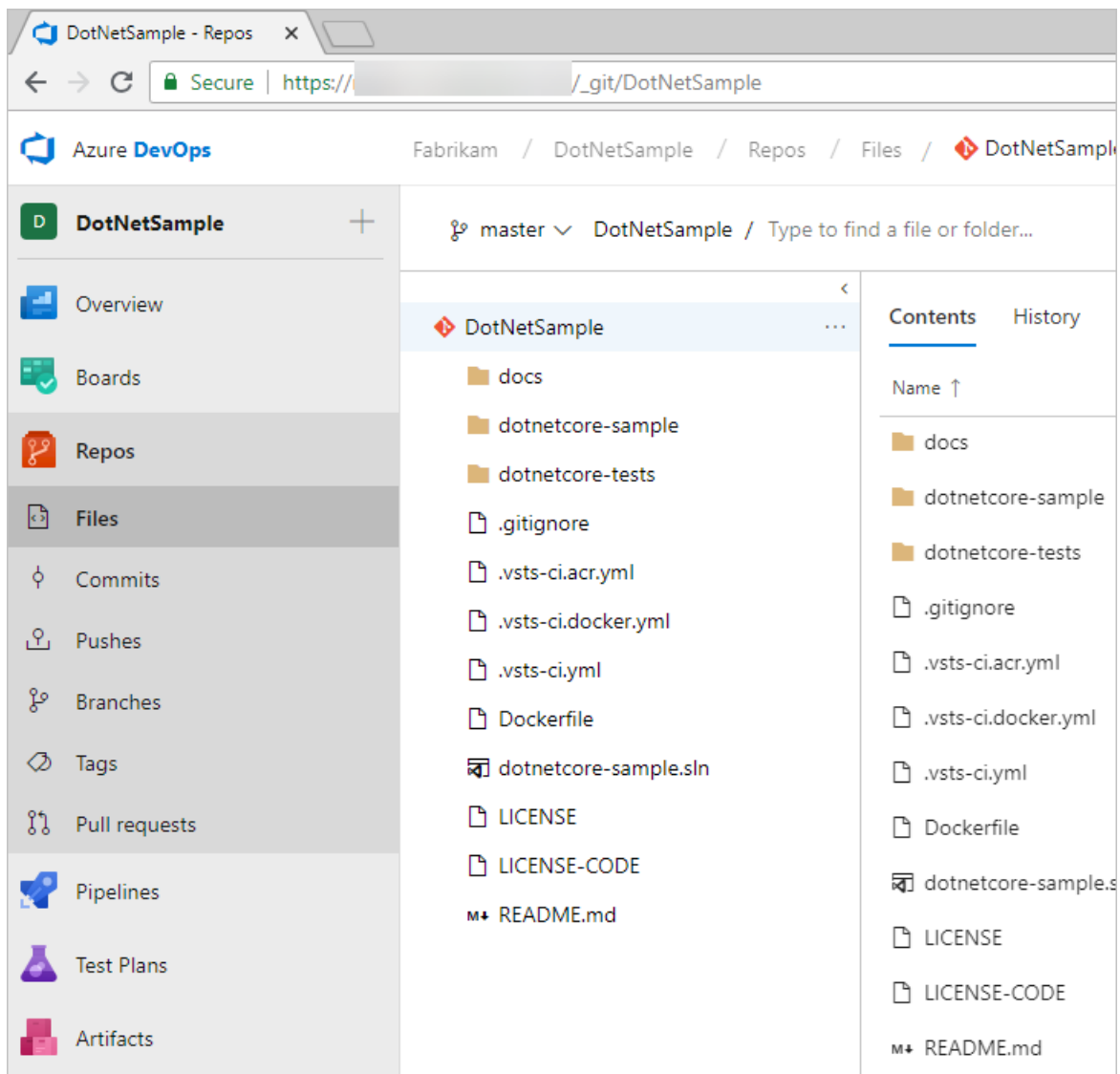
- Add, configure, and manage dashboards
- Configure widgets that you add to dashboards
- Quickly navigate to different areas of your project

To learn more, see [Dashboards](#).

Source control

- [New navigation](#)
- [Previous navigation](#)

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



From Azure Repos for Git you can perform the following tasks:

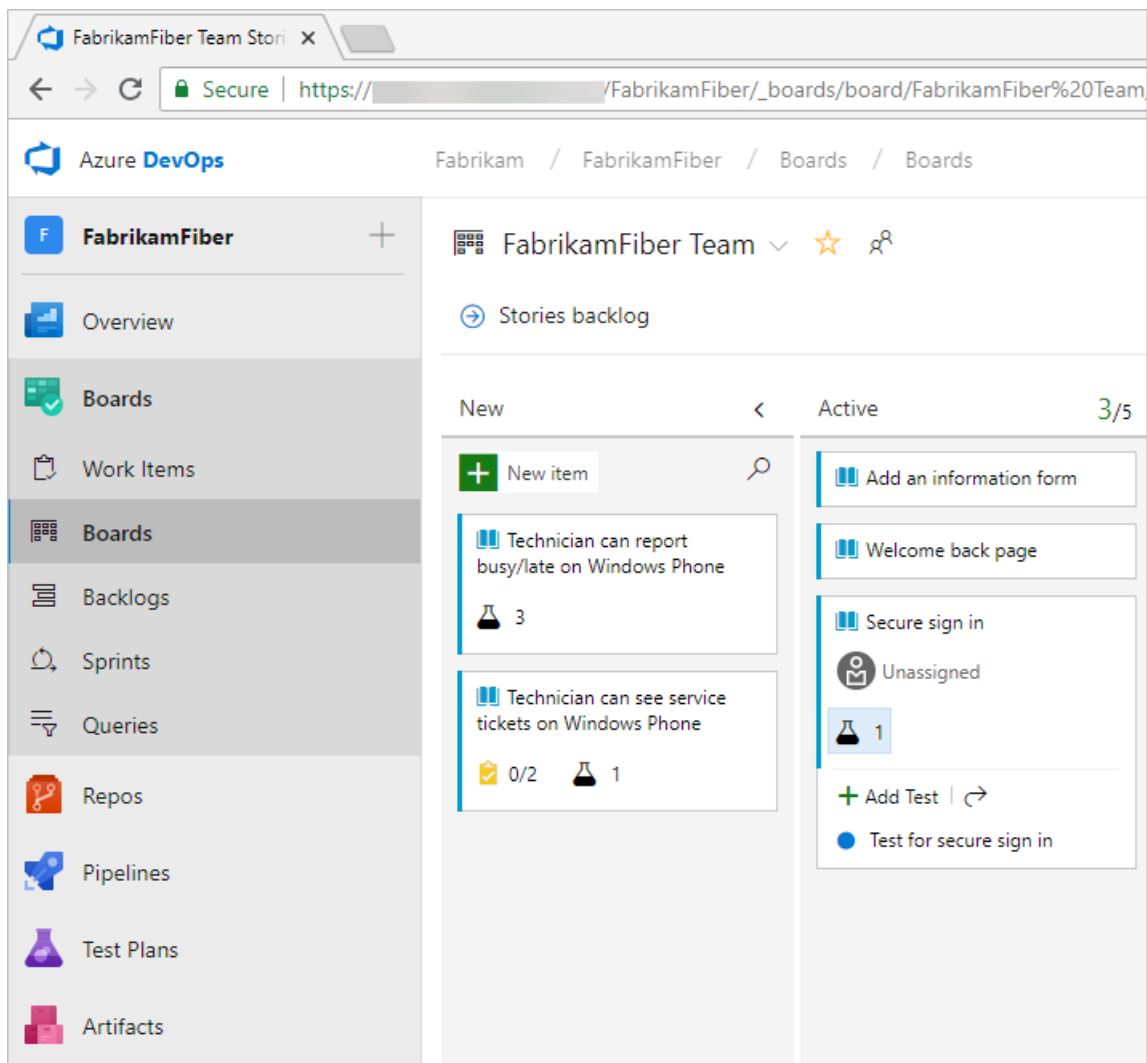
- Review, download, and edit files and review the change history for a file
- Review and manage commits that have been pushed
- Review, create, approve, comment, and complete pull requests
- Add and manage Git tags

To learn more, see the overviews for [Git](#) or [TFVC](#).

Plan and track work

- [New navigation](#)
- [Previous navigation](#)

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



Specifically, you can perform the following tasks:

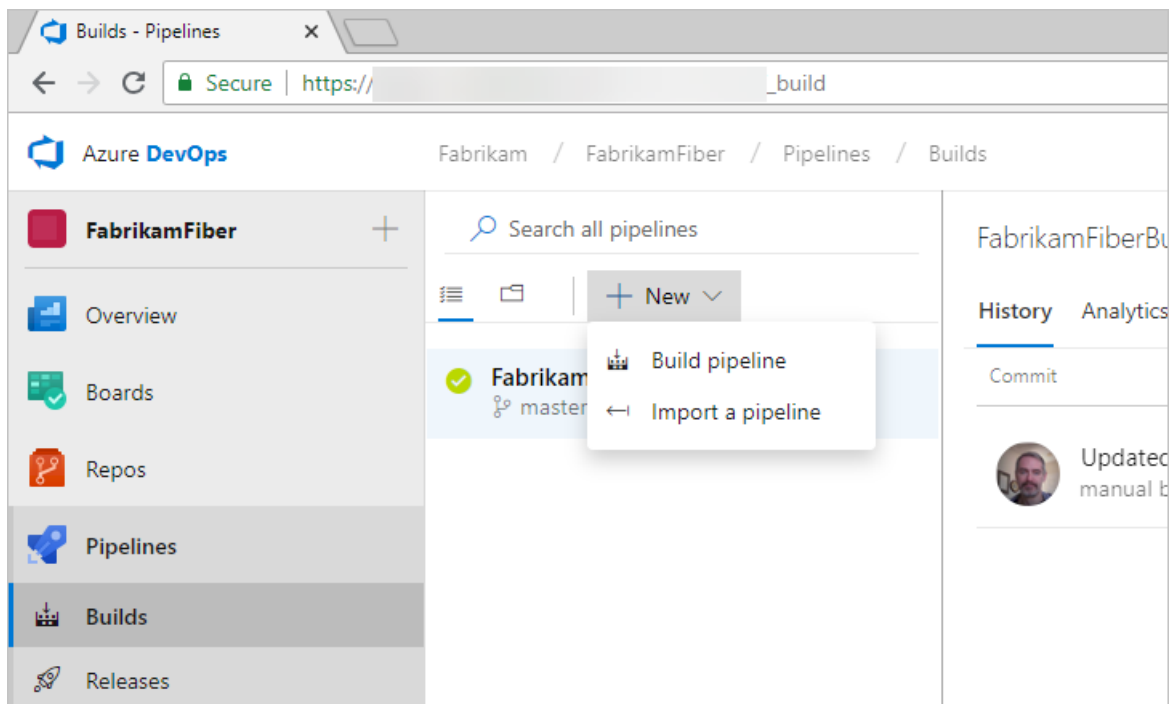
- Add and update work items
- Define work item queries and create status and trend charts based on those queries
- Manage your product backlog
- Plan sprints using sprint backlogs
- Review sprint tasks and update tasks through the task boards
- Visualize the work flow and update status using Kanban boards
- Manage portfolios by grouping stories under features and features under epics

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

Continuous integration and deployment

- [New navigation](#)
- [Previous navigation](#)

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



Use pipelines to implement continuous integration and continuous delivery.

- **Build automation:** Define the steps to take during build and the triggers that will initiate a build.
- **Release management:** Supports a rapid release cadence and management of simultaneous releases. You can configure release pipelines that represent your environments from development to production. Run automation to deploy your app to each environment. Add approvers to sign off that the app has been successfully deployed in an environment. Create your release manually or automatically from a build. Then track your releases as they are deployed to various environments.

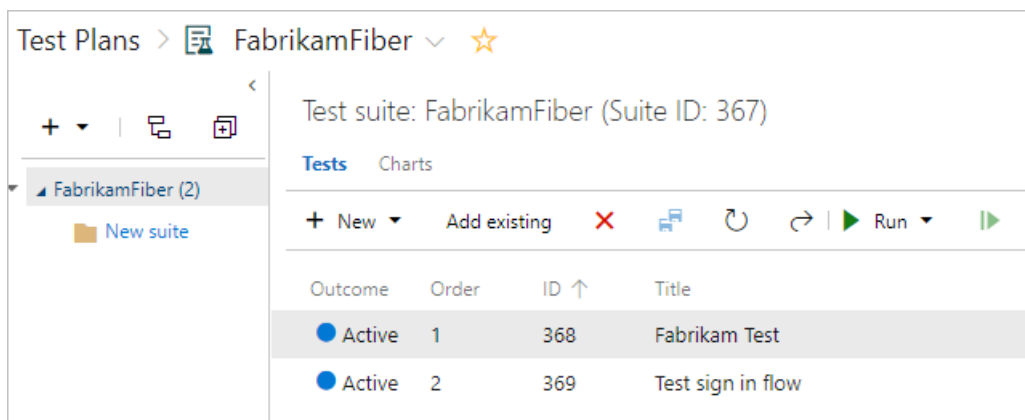
To learn more, see [Continuous integration on any platform](#).

Manual, exploratory, and load testing

Test features support manual and exploratory testing, load or performance testing, and continuous testing.

- [New navigation](#)
- [Previous navigation](#)

Test Plans supports creating and managing manual tests.



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 test case creation
- 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

To learn more, see [Testing overview](#).

Collaboration services

The following services work across the previously mentioned services to support:

- 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
 - Request and manage feedback
 - Analytics service, Analytic 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 project collection
 - Request and manage feedback
 - SQL Server Reporting
-
- Dashboards
 - 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 project collection
 - Request and manage feedback
 - Team (chat) rooms
 - SQL Server Reporting

NOTE

Team rooms are deprecated for TFS 2017.2. Instead, we recommend you [use service hooks to integrate with Microsoft Teams](#).

- Dashboards
 - Linking of work items, commits, pull requests and other artifacts to support traceability
 - Alerts and change notifications managed per user or for teams
 - Request and manage feedback
 - Team (chat) rooms
 - SQL Server Reporting
-
- Team home page
 - Linking of work items, commits, pull requests and other artifacts to support traceability

- Alerts and change notifications managed per user or for teams
- Request and manage feedback
- Team (chat) rooms
- SQL Server Reporting

Service hooks

Service hooks enable you to perform tasks on other services when events happen within your project hosted on Azure DevOps Services or TFS. For example, you can send a push notification to your team's mobile devices when a build fails. Service hooks can also be used in custom apps and services as a more efficient way to drive activities when events happen in your projects.

The following services are available as the target of service hooks. To learn about others apps and services that integrate with our Azure DevOps Services or TFS, visit the [Visual Studio Marketplace](#)

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

Cloud-hosted services based on usage

The following services support your DevOps operations.

- Cloud-based build and deployment Microsoft-hosted agents
- On-premises self-hosted agents to support build and deployment
- Cloud-based performance/load testing lets you load test your code by simulating high traffic

To learn more, see [Pricing](#).

Azure cloud-hosted services

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

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

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

Administrative services

There are a number of features and tasks associated with administering a collaborate software development environment. You perform most of these tasks through the web portal. To learn more, see [About user, team, project, and organization-level settings](#)

- [New navigation](#)
- [Previous navigation](#)

The screenshot shows the Azure DevOps web interface. The browser address bar displays the URL `https://tfs.fabrikam.com/_settings/projectOverview`. The page title is "Project Settings > Overview". On the left, a sidebar lists project components: Overview, Boards, Repos, Pipelines, Test Plans, and Artifacts. A secondary menu on the right lists settings categories: Overview, Services, Teams, Security, Notifications, Service hooks, Dashboards, Work (with sub-items: Project configuration, Team configuration), Build and release (with sub-items: Service connections, Project agent pools, Release retention), and a "Process" section at the bottom. The "Overview" section is active, showing "Project details". It includes a project name "FabrikamFiber" with a "Rename" button, a description "The default project team." in a text box, a visibility dropdown set to "Private" with an "Edit" button, and a "Process" dropdown set to "Scrum".

Related articles

- [Understand differences between Azure DevOps Services and TFS](#)
- [Key concepts](#)
- [Client-server tools](#)
- [Software development roles](#)
- [Pricing](#)

Visual Studio Team Services is now Azure DevOps Services

9/10/2018 • 2 minutes to read • [Edit Online](#)

On September 10th, 2018, Microsoft renamed Visual Studio Team Services (VSTS) to "Azure DevOps Services". For more information about this change, see [this blog post](#).

VSTS features are now separate services:

VSTS FEATURE NAME	AZURE DEVOPS SERVICE NAME	DESCRIPTION
Build & release	Azure Pipelines	CI/CD that works with any language, platform, and cloud.
Code	Azure Repos	Unlimited cloud-hosted private Git and TFVC repos 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, and NuGet package feeds from public and private sources.

Currently, you can acquire only **Azure Pipelines** as a separate service. In the future, you'll be able to acquire each service separately or all together as **Azure DevOps Services**. If you are already a VSTS subscriber, you have access to all of the services now.

NOTE

You can [disable select services from the user interface](#).

Follow the [Azure DevOps release notes](#) to get news on the latest updates!

Can I still use visualstudio.com?

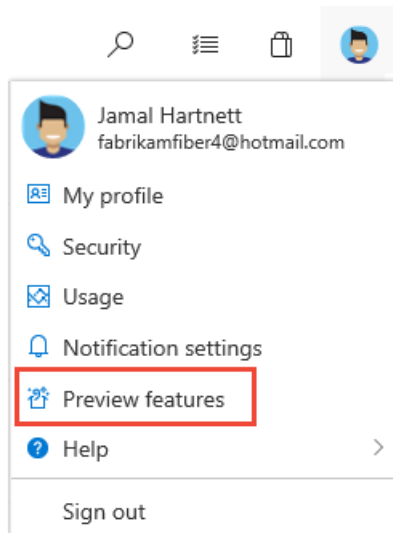
Yes. We've moved to the new `dev.azure.com` domain name as the primary URL for new organizations. (Specifically, `https://dev.azure.com/{your organization}/{your project}`.) If you'd like to change your URL to be based on `dev.azure.com` as the primary, an organization administrator can change this from the organization settings page.

We will continue to post news about this transition on the [Microsoft DevOps blog](#) and in the [Azure DevOps release notes](#).

Can I still use the old interface?

Yes. While Azure DevOps Services features [redesigned navigation](#), you can still use the previous user interface by

choosing your profile icon and selecting **Preview features** from the drop-down menu.



Then, toggle the **New Navigation** option to **Off**.



Preview features

The following preview features are available for your evaluation. Help us make them better!

for me [Jamal Hartnett] ▼

New build result page ☐ On

Lights up new build results page.

New builds hub ☒ On

Lights up new builds hub.

New Navigation ☐ Off

Turns on the new global navigation experience. [Learn more](#)

New release progress views ☒ On

Turn on the new release views to visualize the progress of your deployment pipelines. [Learn more](#)

New Releases Hub ☒ On

New experience to create folders and manage release pipelines.

Turning off New Navigation will also change the Azure DevOps Services names to the corresponding old VSTS feature names, such as **Work** instead of **Boards**.

To return to the Azure DevOps service-oriented UI, choose your profile icon, select **Preview features**, and re-enable **New Navigation**.

What about Team Foundation Server (TFS)?

As of September 10, 2018, [Team Foundation Server](#) (TFS) is unchanged and fully-supported as our on-premises

Agile workflow and DevOps product. The latest version is TFS 2018; the latest servicing update is **Team Foundation Server 2018 Update 3**, which released on **August 28, 2018**.

Quickstart: Code with Git

9/10/2018 • 3 minutes to read • [Edit Online](#)

Azure DevOps Services

In this quickstart, you learn how to share your code with others. After you create a new organization and project in Azure DevOps, you can begin coding with Git.

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

Install Git command line tools

1. Install one of the following Git command line tools:

- To install Git for Windows, including Git Credential Manager, see [Install the Git Credential Manager - Windows](#)
- To install Git for macOS and Linux, see [Install the Git Credential Manager - macOS and Linux](#)

Clone the repo to your computer

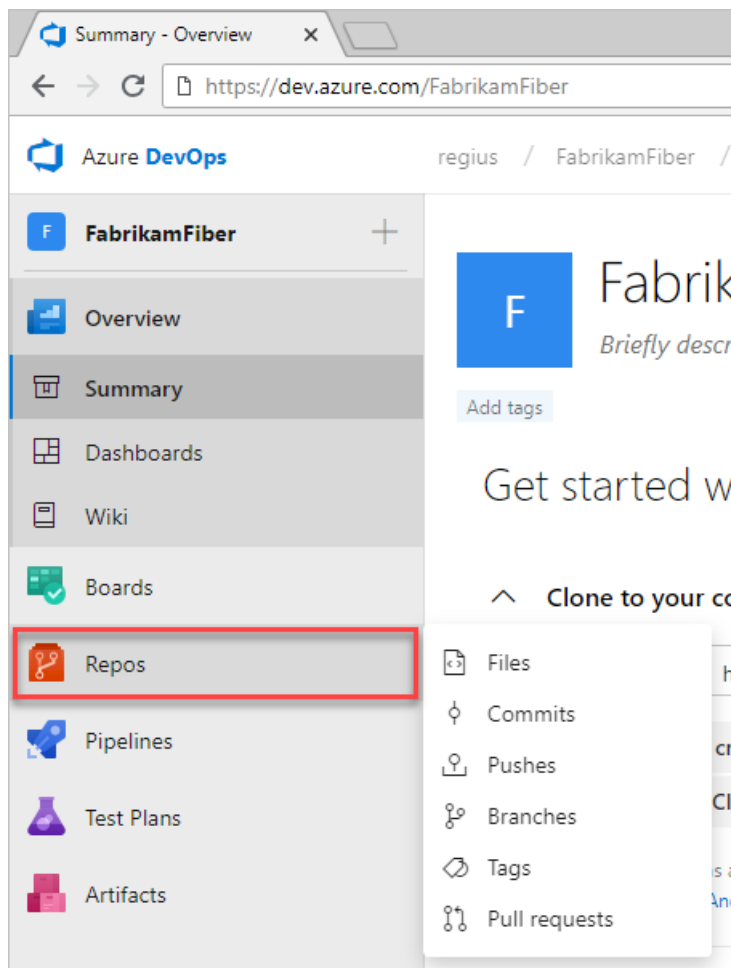
NOTE

The **New navigation** feature provides a vertical navigation experience and is in preview for Azure DevOps Services. When you [enable new navigation](#), you automatically enable several new Agile tool features that are described in the [New Work Hubs](#) blog post.

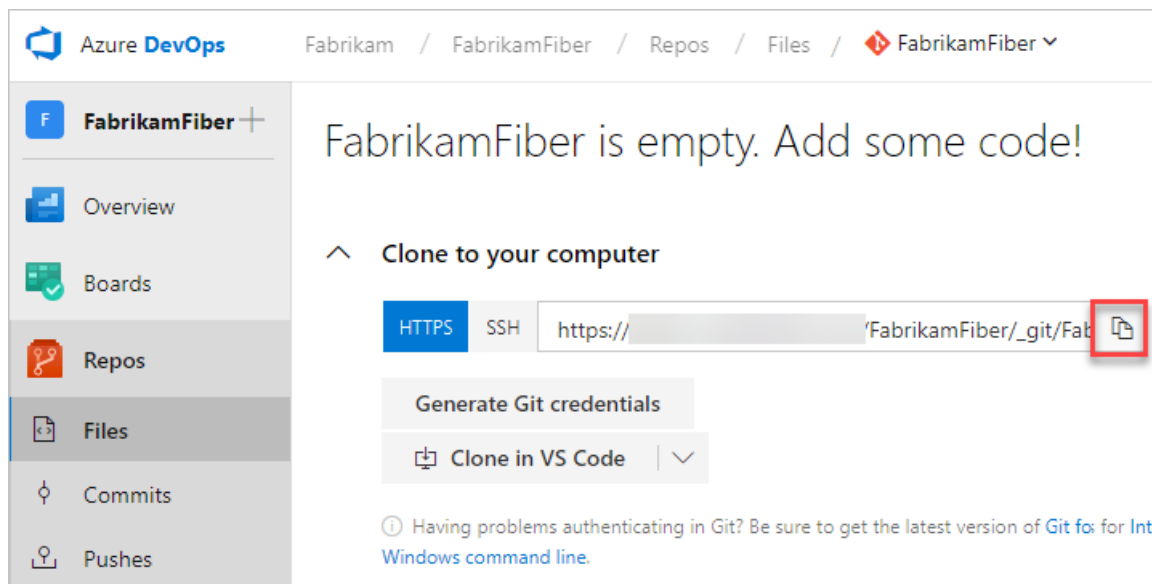
On-premises Microsoft Team Foundation Server users can select **Previous navigation** for guidance.

- [New navigation](#)
- [Previous navigation](#)

1. From your web browser, open the project for your Azure DevOps organization and select **Repos**. If you don't have a project, [create one now](#).



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



1. Open the Git command window (Git Bash on Git for Windows), navigate 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 the **Clone URL** in the previous step, as shown in the following example.

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

A copy of the code is downloaded in Git, including all [commits](#) and [branches](#) from the repo, into a new folder for you to work with.

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

Work with the code

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

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

```
cd contoso-demo
```

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

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

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

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

```
git push
```

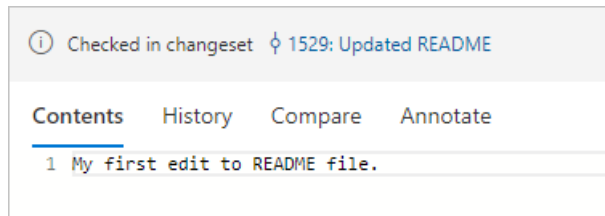
View history

- [New navigation](#)
- [Previous navigation](#)

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 web portal interface. At the top, the breadcrumb navigation is 'FabrikamFiber / FabrikamFiber / Code / Files / \$/FabrikamFiber'. Below this, the current path is '\$/FabrikamFiber /'. On the left, a file explorer shows the repository structure with folders like 'billing-service', 'BuildProcessTemplates', 'Drops', 'Hello World', 'TeamBuildTypes', 'Tools' and files like '.tfignore' and 'README'. On the right, the 'History' tab is selected and highlighted with a red box. The history list shows two commits: one on 'Wednesday, August 8, 2018' titled 'Added file README' by 'Jamal Hartnett', and another on 'Tuesday, October 13, 2015' titled '***NO_CI***' by 'Elastic Build (FabrikamFiber)'. The 'History' tab is the active view, showing a list of commits with their titles, authors, and dates.

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



Next steps

[Set up continuous integration & delivery](#) or [Plan & track work](#)

Or, [learn more about working with a Git repo](#)

Use the visual designer

9/10/2018 • 16 minutes to read • [Edit Online](#)

NOTE

Build and release *pipelines* are called *definitions* in TFS 2018 and in older versions. *Service connections* are called *service endpoints* in TFS 2018 and in older versions.

NOTE

This guidance applies to TFS version 2017.3 and newer.

TIP

For build pipelines, we recommend that you use YAML instead of the visual designer that is explained below. YAML allows you to use the same branching and code review practices for your pipeline as you would for your application code. See [Create your first pipeline](#).

We'll show you how to use the visual designer in Azure Pipelines to create a build and release that prints "Hello world". If you plan to use a YAML file instead of the visual designer, then see [Create your first pipeline](#) instead.

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

Prerequisites

- You need an Azure DevOps organization. If you don't have one, you can [create one for free](#). If your team already has one, then make sure you're an administrator of the project you want to use.
- A [self-hosted Windows agent](#).

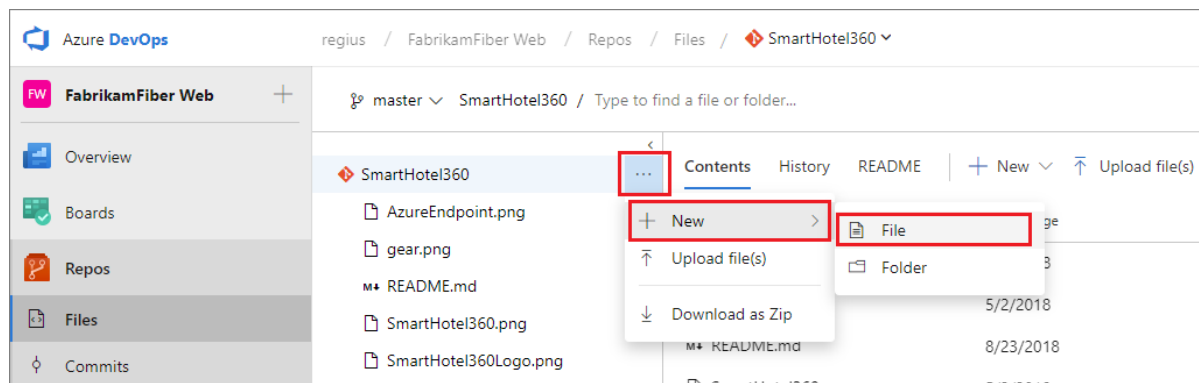
Add a script to your repository

Create a PowerShell script that prints `Hello world`.

1. Go to **Azure Repos**.

2. Add a file.

- [New navigation](#)
- [Previous navigation](#)



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

HelloWorld.ps1

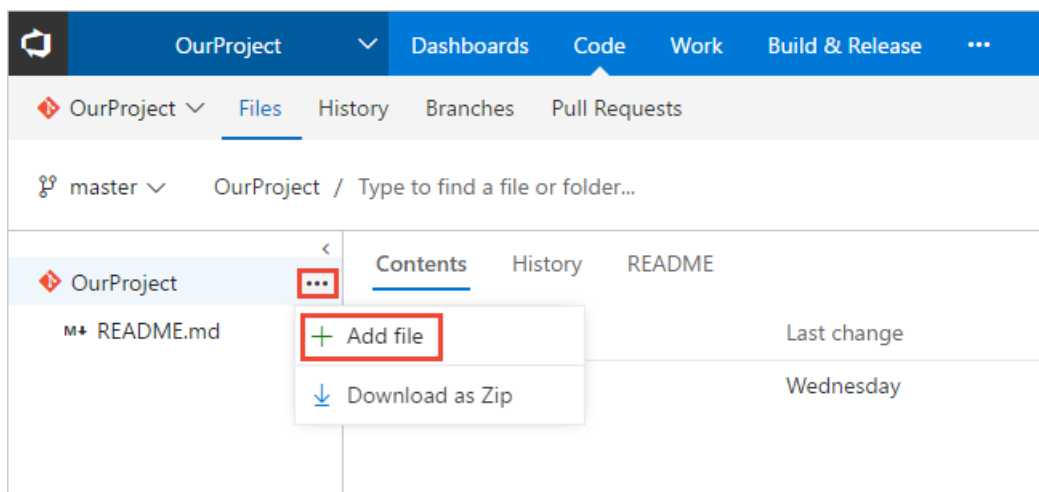
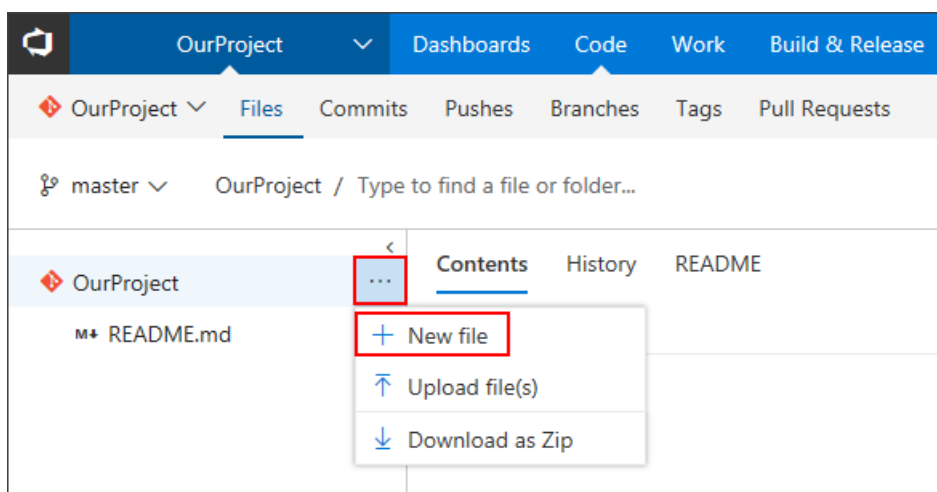
4. Copy and paste this script.

Write-Host "Hello world"

5. **Commit** (save) the file.

1. Go to the **Code** hub.

2. Add a file.



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.

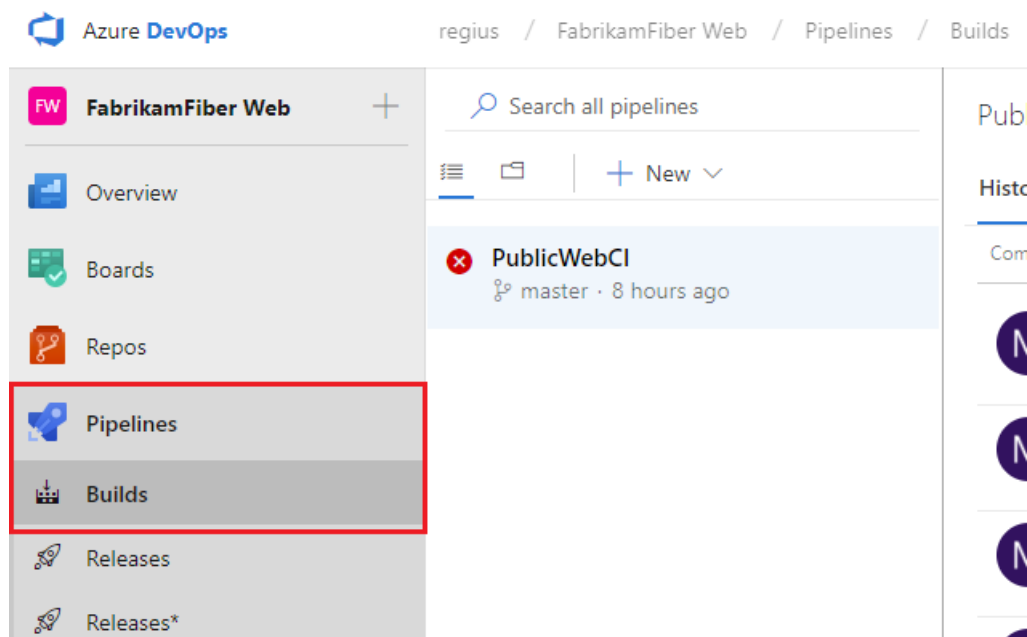
When you're ready to begin building and deploying a real app, you can use a wide range of version control clients and services with Azure Pipelines CI builds. [Learn more](#).

Create a build pipeline

Create a build pipeline that prints "Hello world."

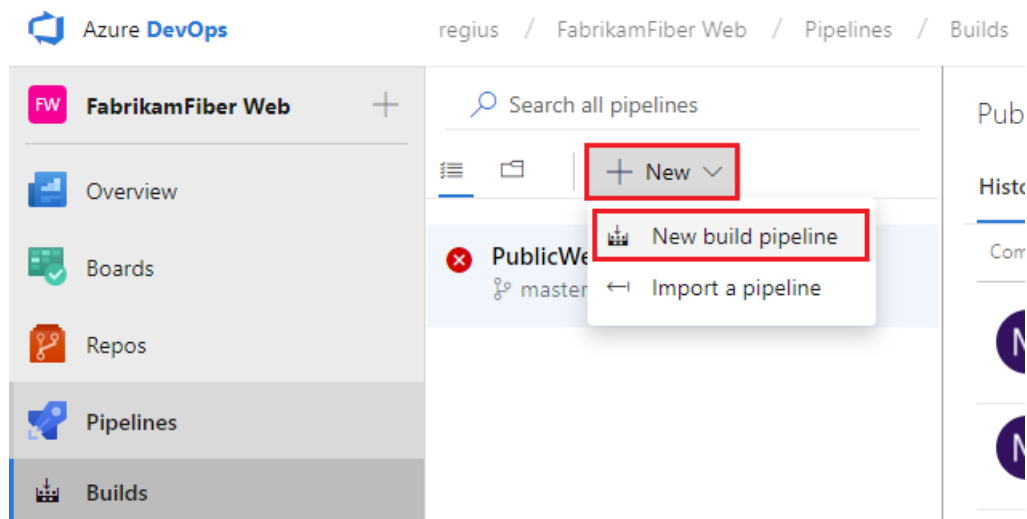
1. Select **Azure Pipelines**, it should automatically take you to the **Builds** page.

- [New navigation](#)
- [Previous navigation](#)



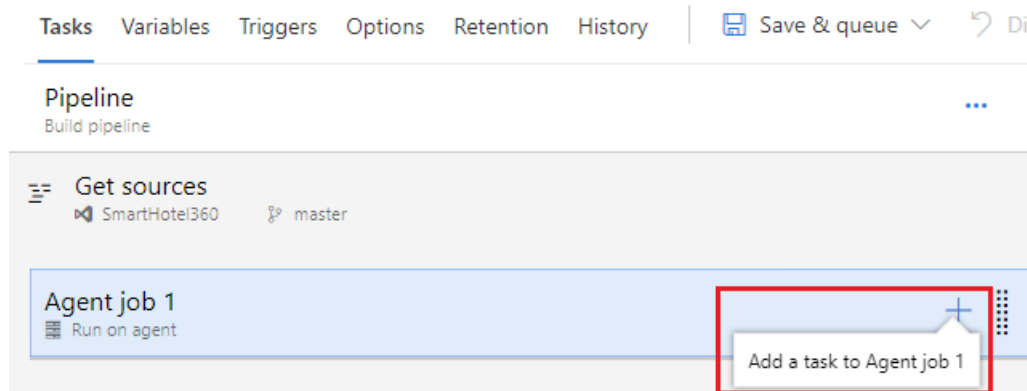
2. Create a new pipeline.

- [New navigation](#)
- [Previous navigation](#)



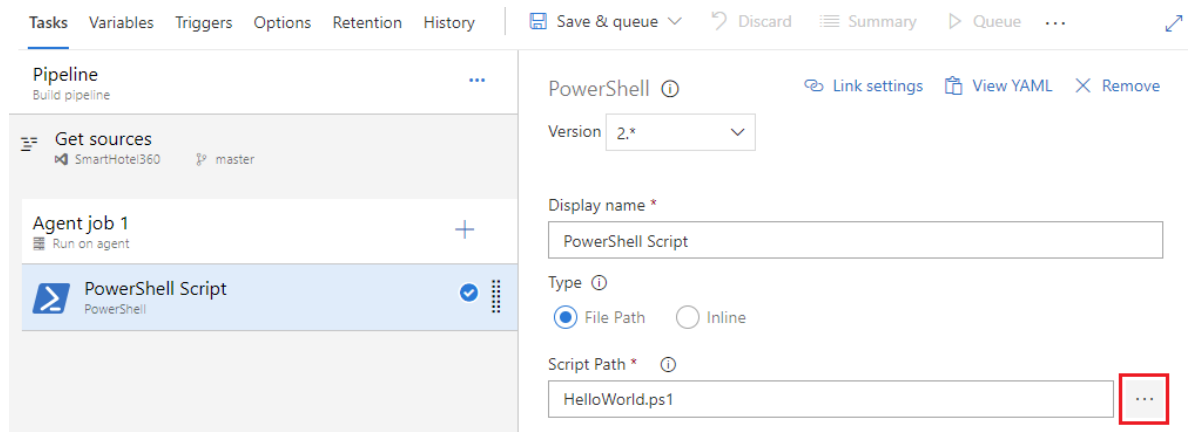
3. Make sure that the **source**, **project**, **repository**, and default **branch** match the location in which you created the script.
4. Start with an **Empty job**.
5. On the left side, select **Pipeline** and specify whatever **Name** you want to use. For the **Agent pool**, select **Hosted VS2017**.
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**.

- [New navigation](#)
- [Previous navigation](#)



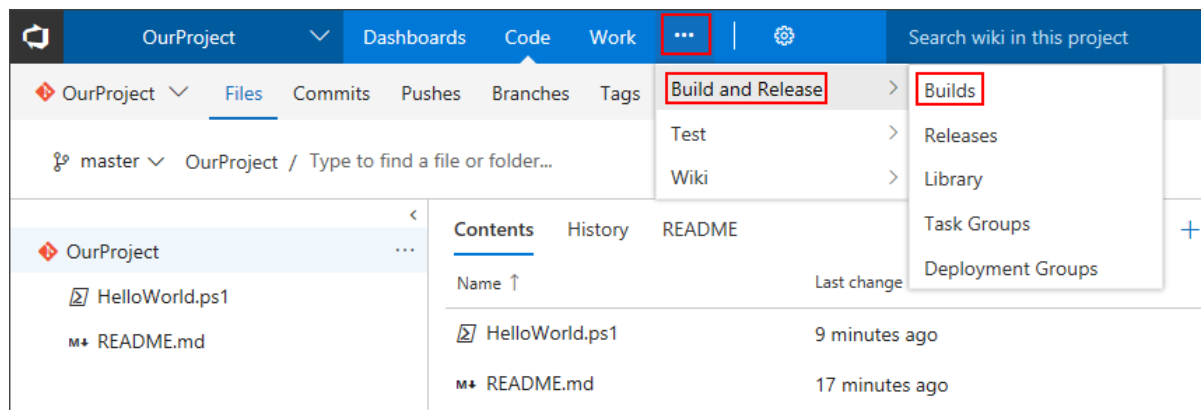
7. On the left side, select your new **PowerShell** script task.
8. For the **Script Path** argument, select the **...** button to browse your repository and select the script you created.

- [New navigation](#)
- [Previous navigation](#)

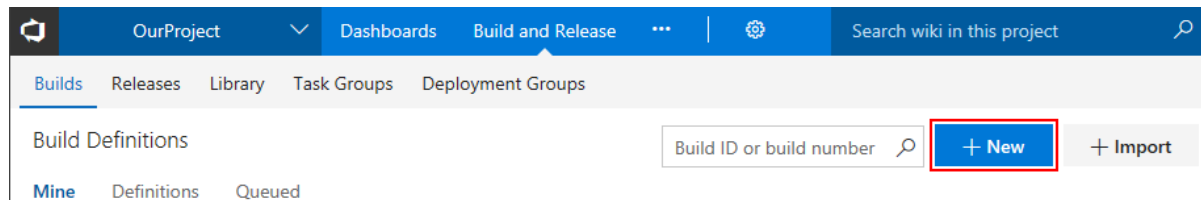


9. Select **Save & queue**, and then select **Save**.

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



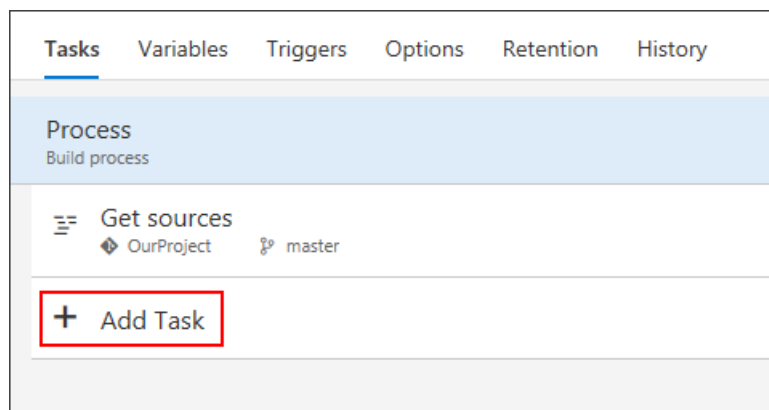
2. Create a new pipeline.



3. Start with an **empty pipeline**

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

5. 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**.



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

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

created.

The screenshot shows the configuration for a PowerShell Script task. On the left, a sidebar lists tasks: 'Get sources' (selected), 'PowerShell Script' (checked), and 'Add Task'. The main area is titled 'PowerShell' and includes fields for 'Version' (1.*), 'Display name' (PowerShell Script), 'Type' (File Path), and 'Script Path' (HelloWorld.ps1). A red box highlights the three-dot menu icon next to the Script Path field.

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

1. Select **Azure Pipelines**, and then the **Builds** tab.

The screenshot shows the Azure DevOps interface. The top navigation bar includes 'OurProject', 'Dashboards', 'Code', and a menu icon. The 'Builds' tab is selected, and a dropdown menu shows options: 'Builds', 'Releases', 'Packages', 'Library', 'Task Groups', 'Deployment Groups*', and 'Explorer'. A red box highlights the 'Builds' option in the dropdown.

2. Create a new pipeline.

The screenshot shows the 'Build Definitions' page in Azure DevOps. The top navigation bar includes 'OurProject', 'Build & Release', and a menu icon. The 'Builds' tab is selected, and a dropdown menu shows options: 'Builds', 'Releases', 'Packages', 'Library', 'Task Groups', 'Deployment Groups*', and 'Explorer'. A red box highlights the '+ New' button.

3. Start with an **empty pipeline**.

4. Select **Pipeline** and specify whatever **Name** you want to use.

5. On the **Options** tab, select **Default** for the **Agent pool**, or select whichever pool you want to use that has Windows build agents.

6. On the **Tasks** tab, make sure that **Get sources** is set with the **Repository** and **Branch** in which you created the script.

7. On the left side select **Add Task**, and then on the right side select the **Utility** category, select the **PowerShell** task, and then select **Add**.

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

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

10. 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**.

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

Artifact name: Enter .

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 'Publish Build Artifacts' task configuration in Azure DevOps. The 'Tasks' tab is active, and the 'Publish Artifact: drop' task is selected. The configuration panel on the right shows the following fields:

- Path to Publish**: HelloWorld.ps1
- Artifact Name**: drop
- Artifact Type**: Server

A red box highlights the 'Path to Publish', 'Artifact Name', and 'Artifact Type' fields.

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

Artifact Name: Enter .

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.

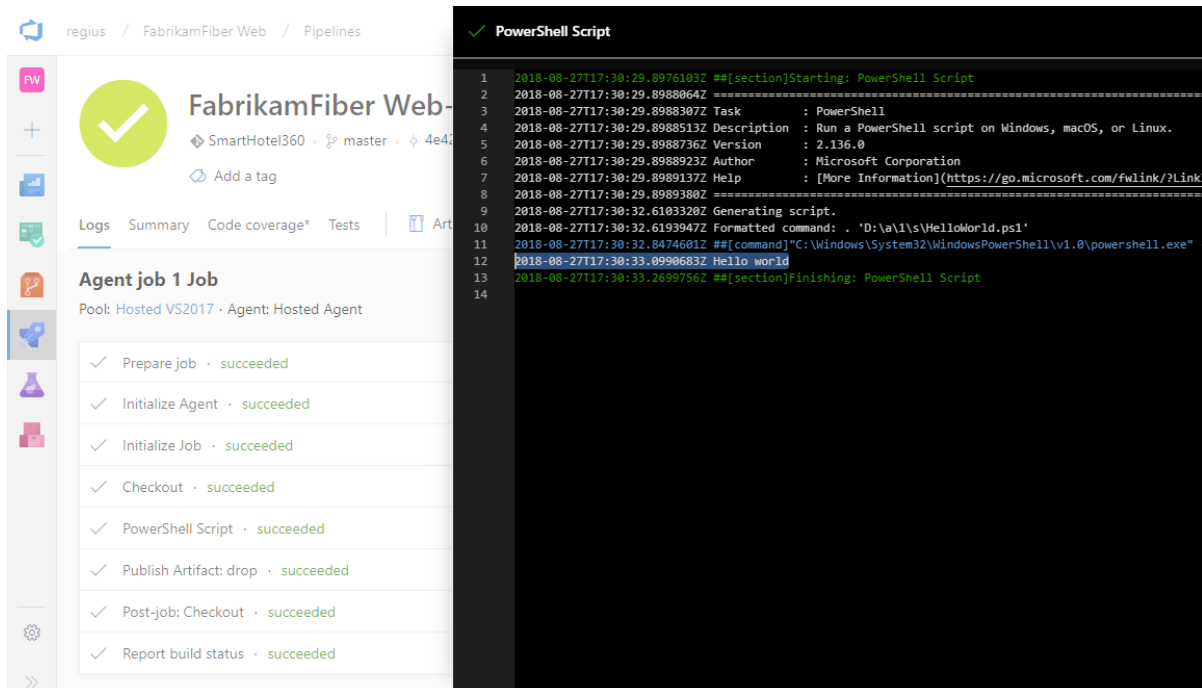
- [New navigation](#)

- [Previous navigation](#)



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.

- [New navigation](#)
- [Previous navigation](#)



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

- [New navigation](#)
- [Previous navigation](#)

regius / FabrikamFiber Web / Pipelines

FW
+
[Icons]

FabrikamFiber Web-CI 162

SmartHotel360 · master · 68131ea : Updated HelloWorld.ps1 · Manual build · R

Add a tag

Logs **Summary** Code coverage* Tests | Artifacts Release Edit Queue

Progression

Deployments
0
No deployments were found for this build.

Build artifacts published ^
1

drop
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.

Builds Releases Library Task Groups Deployment Groups

... > OurProject-CI

Build #7 has been queued.

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 'Builds' tab in Azure Pipelines. On the left, a list of build steps is shown with green checkmarks: Initialize Agent, Initialize Job, Get Sources, PowerShell Script, Publish Artifact: drop, Post Job Cleanup, Finalize build, and Report build status. The main area displays 'Hello world / Build 1722 / Build' with a toggle switch set to 'Build not retained'. Below this, a green banner indicates 'Build succeeded'. A progress bar shows the build ran for 4 seconds (Hosted Agent), completed 1 seconds ago. The 'Console' tab is selected, showing the following output:

```
git checkout -b <new-branch-name>
HEAD is now at 0ab86c0... Updated HelloWorld.ps1
Finishing: Get Sources
Starting: PowerShell Script
Task       : PowerShell
Description: Run a PowerShell script
Version    : 1.2.3
Author     : Microsoft Corporation
Help       : [More Information](https://go.microsoft.com/fwlink/?LinkID=613736)
'd:\a\1\s\HelloWorld.ps1'
Hello world
```

4. Go to the build summary.

This screenshot shows the 'Builds' tab with 'Build 1722' selected. The 'Build succeeded' banner is prominent. The breadcrumb navigation shows 'Hello world / Build 1722 / Build'. The 'Build succeeded' banner is green and contains the text 'Build succeeded'.

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

This screenshot shows the 'Builds' tab with 'Build 1722' selected. The 'Artifacts' tab is selected, showing a list of artifacts. The artifact 'drop' is listed with a 'Download' button and an 'Explore' button. The 'Artifacts Explorer' dialog is open, showing the 'drop' artifact and the file '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.

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

This queues a new build on the agent. 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.

Builds Releases Packages Library Task Groups Deployment Groups* Explorer

✓ Build 1722

✓ Build

✓ Initialize Agent

✓ Initialize Job

✓ Get Sources

✓ PowerShell Script

✓ Publish Artifact: drop

✓ Post Job Cleanup

✓ Finalize build

✓ Report build status

Hello world / Build 1722 / Build ☐ Build not retained

Edit build definition Queue new build... Download all logs as zip Release

Build succeeded

Build

Ran for 4 seconds (Hosted Agent), completed 1 seconds ago

Console Logs Code coverage* Tests

```
git checkout -b <new-branch-name>
HEAD is now at 0ab86c0... Updated HelloWorld.ps1
*****
Finishing: Get Sources
*****
Starting: PowerShell Script
*****
Task       : PowerShell
Description: Run a PowerShell script
Version    : 1.2.3
Author     : Microsoft Corporation
Help       : [More Information](https://go.microsoft.com/fwlink/?LinkID=613736)
*****
. 'd:\a\1\s\HelloWorld.ps1'
Hello world
```

3. Go to the build summary.

Builds Releases Packages Library Task Groups Deployment Groups*

✓ Build 1722

✓ Build

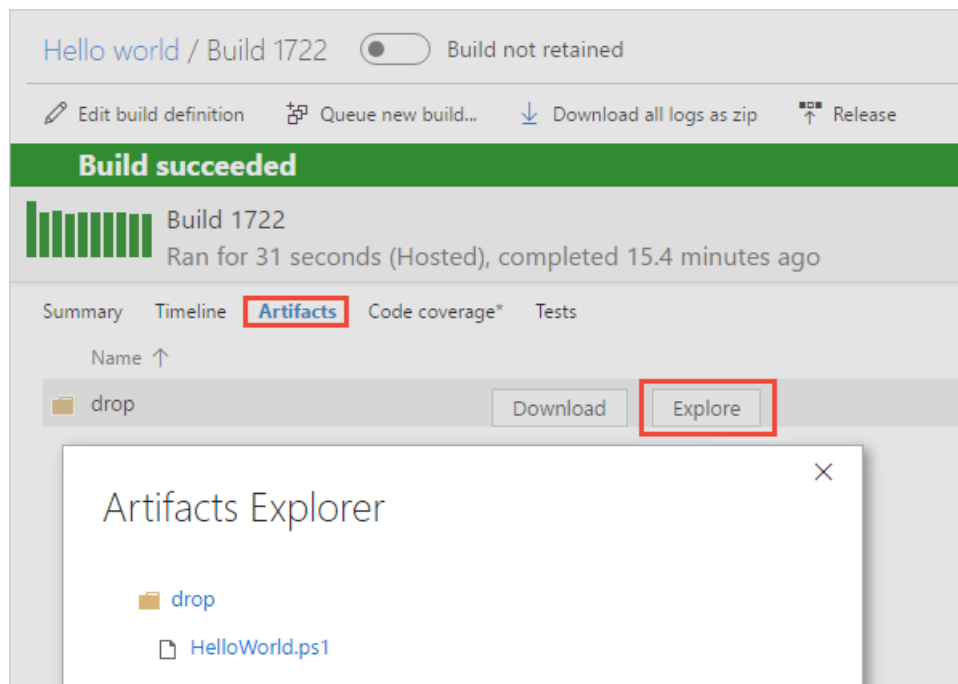
✓ Initialize Agent

Hello world / **Build 1722** / Build ☐

Edit build definition Queue new build...

Build succeeded

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



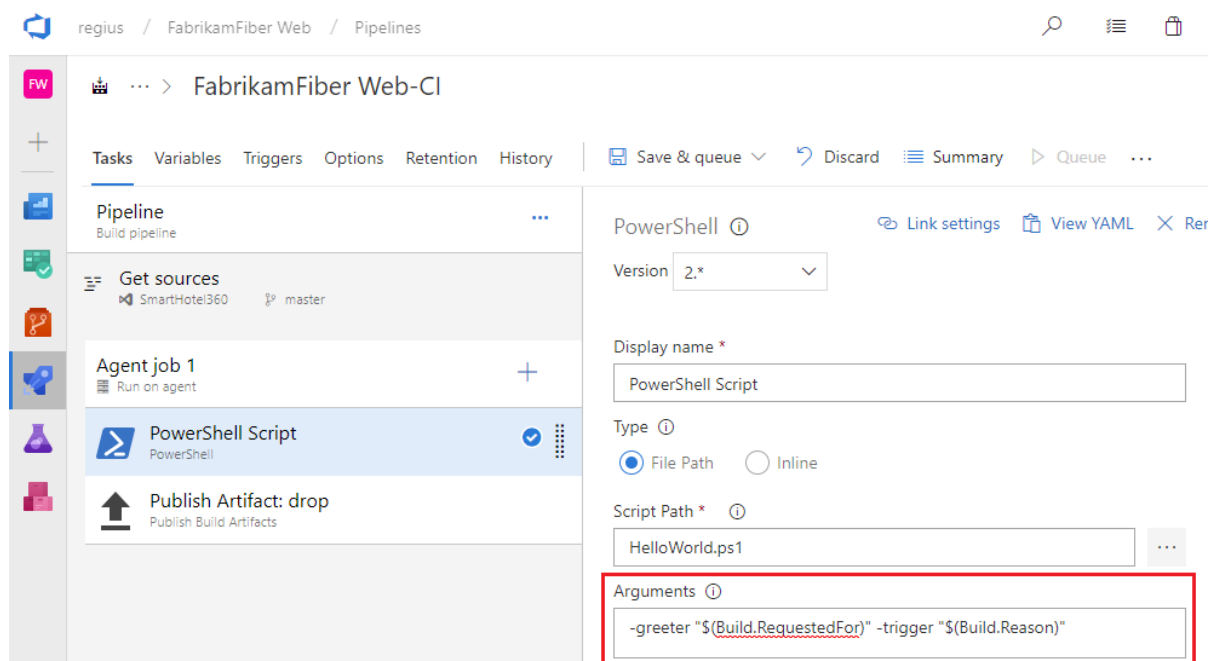
You can view a summary of all the builds or drill into the logs for each build at any time by navigating to the **Builds** tab in **Build and Release**. For each build, you can also view a list of commits that were built and the work items associated with each commit. You can also run tests in each build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

- [New navigation](#)
- [Previous navigation](#)



Tasks Variables Triggers Options Retention History

Process
Build process

Get sources
OurProject master

PowerShell Script
PowerShell

Publish Artifact: drop
Publish Build Artifacts

+ Add Task

PowerShell ⓘ
Version 1.*
Display name
PowerShell Script
Type ⓘ *
File Path
Script Path ⓘ *
HelloWorld.ps1
Arguments ⓘ
-greeter "\$(Build.RequestedFor)" -trigger "\$(Build.Reason)"
Advanced
Control Options

Link settings

Arguments

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

- Save the build pipeline.
- Go to the **Code** hub, **Files** tab.
- Select the **HelloWorld.ps1** file, and then **Edit** the file.
- Change the script as follows:

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

- Commit** (save) the script.
- 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.
- 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.
- Select the new build that was created and view its log.
- 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.

- [New navigation](#)
- [Previous navigation](#)

regius / FabrikamFiber

✓ PowerShell Script

```

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 2018-08-30T17:33:29.1730205Z Author : Microsoft Corporation
7 2018-08-30T17:33:29.1730373Z Help : [More Information](https://go.microsoft.com/fwlink/?LinkID=
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" -No
12 2018-08-30T17:33:31.3101400Z 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

```

Agent job 1 Job
Pool: Hosted VS2017

✓ Build 1723

- ✓ Build
- ✓ Initialize Agent
- ✓ Initialize Job
- ✓ Get Sources
- ✓ PowerShell Script
- ✓ Publish Artifact: drop
- ✓ Post Job Cleanup
- ✓ Finalize build
- ✓ Report build status

Hello world / Build 1723 / Build / PowerShell Script ☐ Build

Edit build definition Queue new build... Download all logs as zip

Build succeeded

PowerShell Script

Ran for 1 seconds (Hosted Agent), completed 74 seconds

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://
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 just 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.

- [New navigation](#)
- [Previous navigation](#)

[All pipelines](#) > **New release pipeline** Save

Pipeline Tasks Variables Retention Options History

Artifacts | + Add

_FabrikamFiber Web-CI

Schedule not set

Stages | + Add

QA

1 job, 0 task

Continuous deployment trigger

Build: _FabrikamFiber Web-CI

☒ Enabled

Creates a release every time a new build is available.

Build branch filters ⓘ

No filters added.

+ Add

Pull request trigger

Build: _FabrikamFiber Web-CI

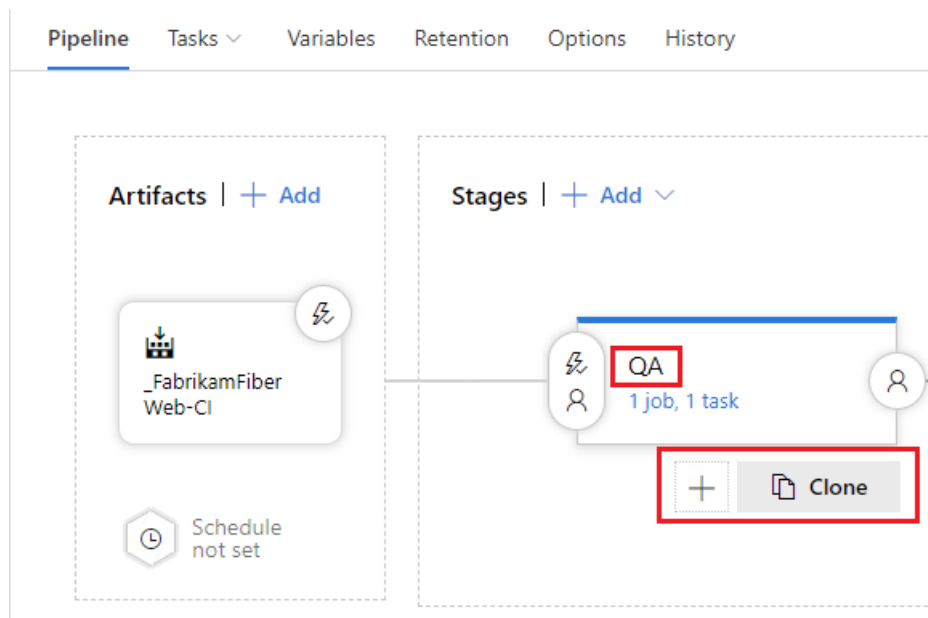
☐ Disabled

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**.

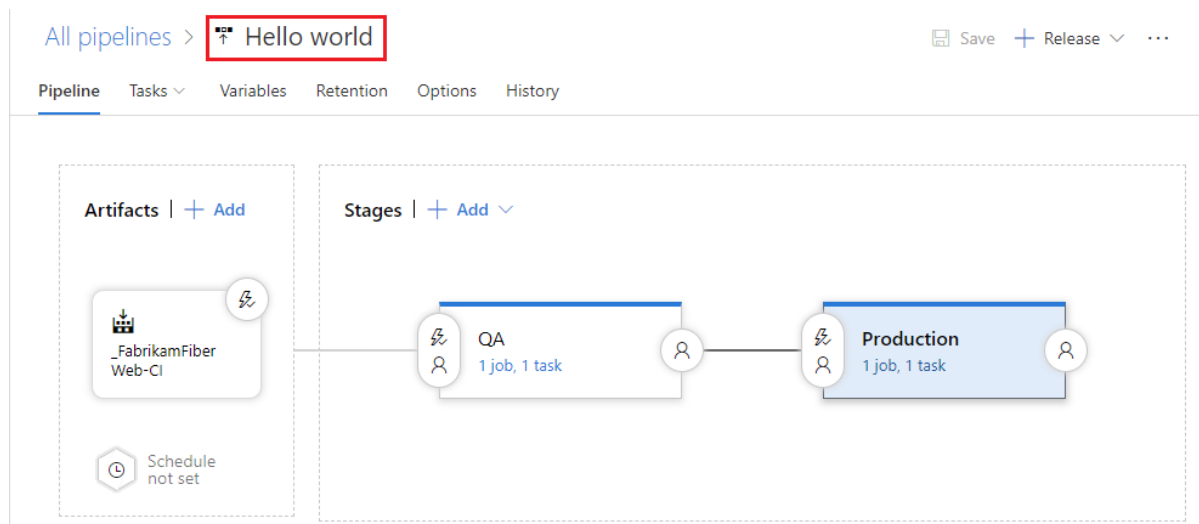
- [New navigation](#)
- [Previous navigation](#)



14. Rename the cloned stage **Production**.

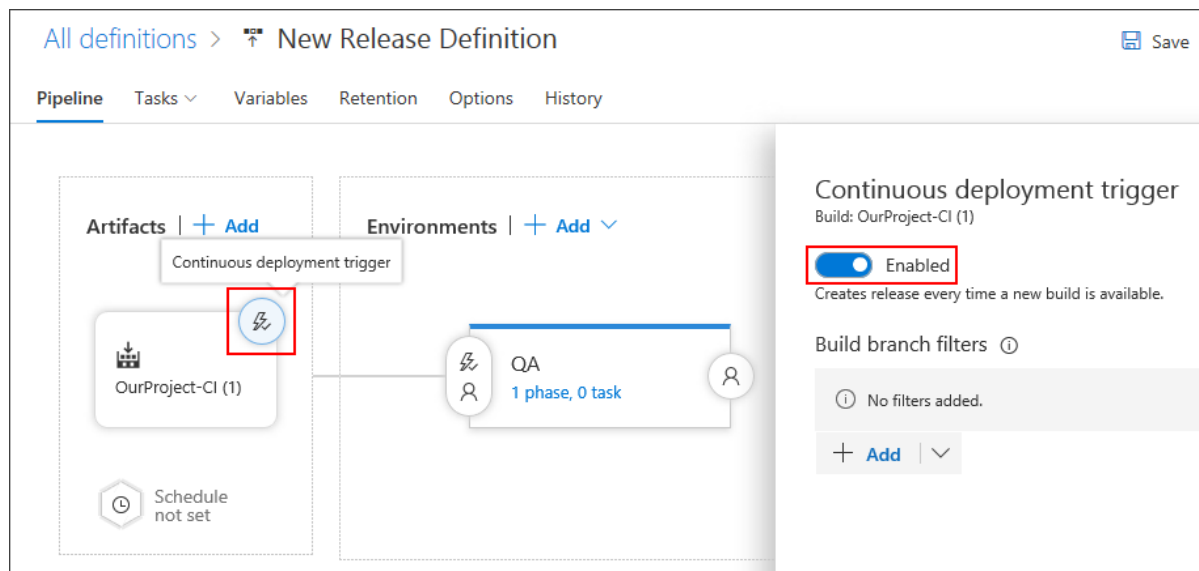
15. Rename the release pipeline **Hello world**.

- [New navigation](#)
- [Previous navigation](#)



16. Save the release pipeline.

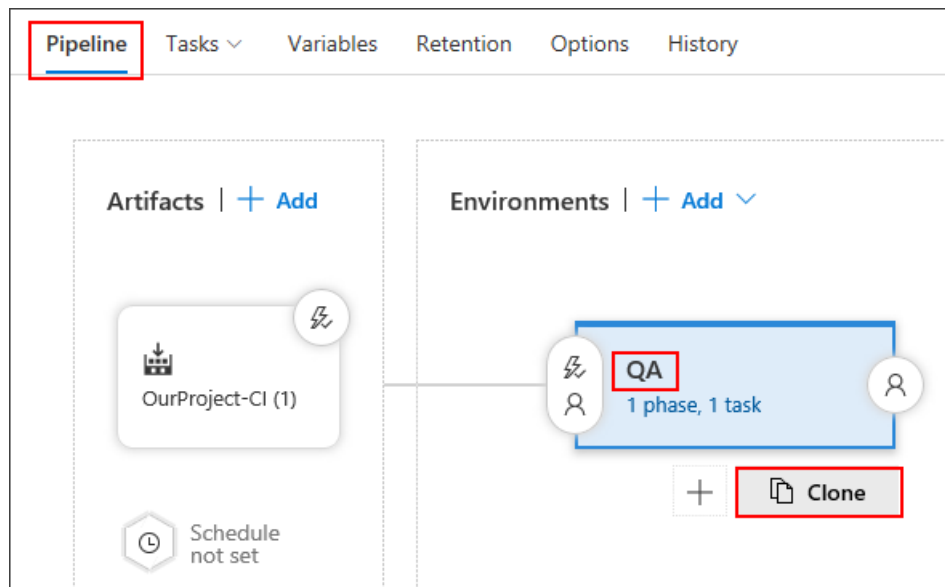
1. Go to the **Build and Release** tab, and then select **Releases**.
2. Select the action to create a **New pipeline**. If a release pipeline is already created, select the plus sign (+) and then select **Create a release definition**.
3. Select the action to start with an **Empty definition**.
4. Name 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.



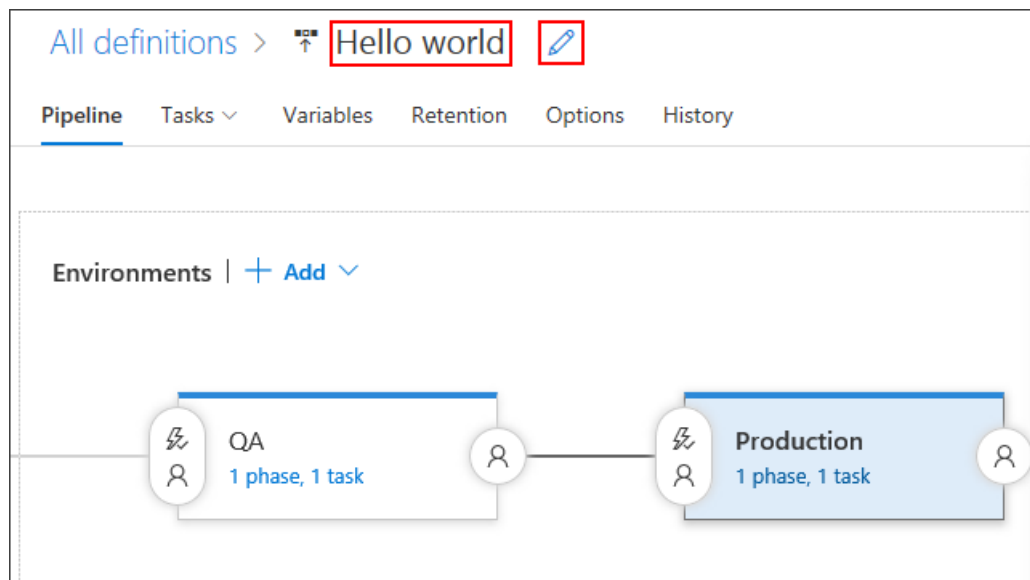
7. Select the **Tasks** tab and select your **QA** stage.
8. Select the plus sign (+) for the job to add a task to the job.
9. On the **Add tasks** dialog box, select **Utility**, locate the **PowerShell** task, and then select its **Add** button.
10. On the left side, select your new **PowerShell**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

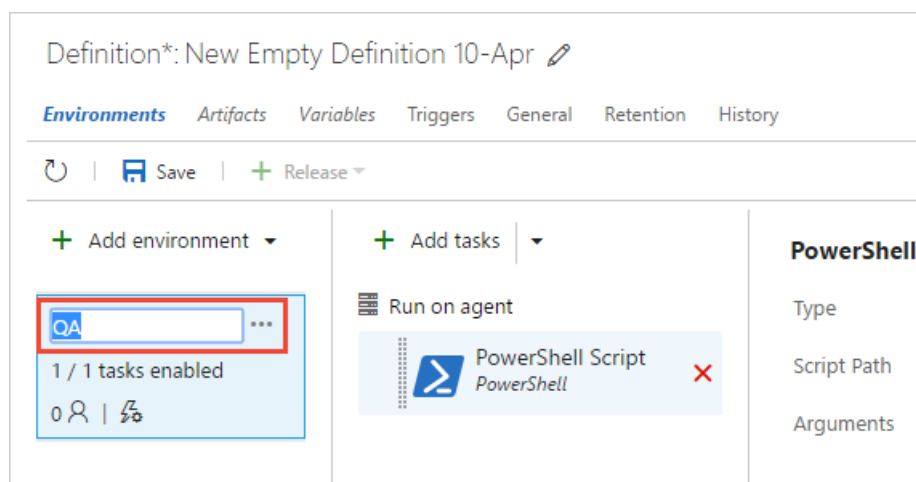


16. Save the release pipeline.

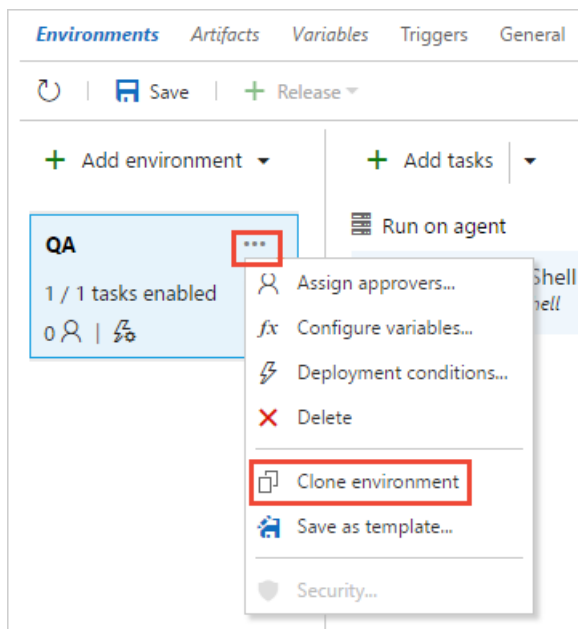
1. Go to **Azure Pipelines**, and then to the **Releases** tab.
2. Select the action to create a **New pipeline**.
3. On the dialog box, select the **Empty** template and select **Next**.
4. Make sure that your **Hello world** build pipeline that you created above is selected. Select **Continuous deployment**, and then select **Create**.
5. Select **Add tasks** in the stage.
6. On the **Task catalog** dialog box, select **Utility**, locate the **PowerShell** task, and then select its **Add** button. Select the **Close** button.
7. For the **Script Path** argument, select the **...** button to browse your artifacts and select the script you created.
8. Add these **Arguments**:

```
-greeter "${Release.RequestedFor}" -trigger "${Build.DefinitionName}"
```

9. Rename the stage **QA**.



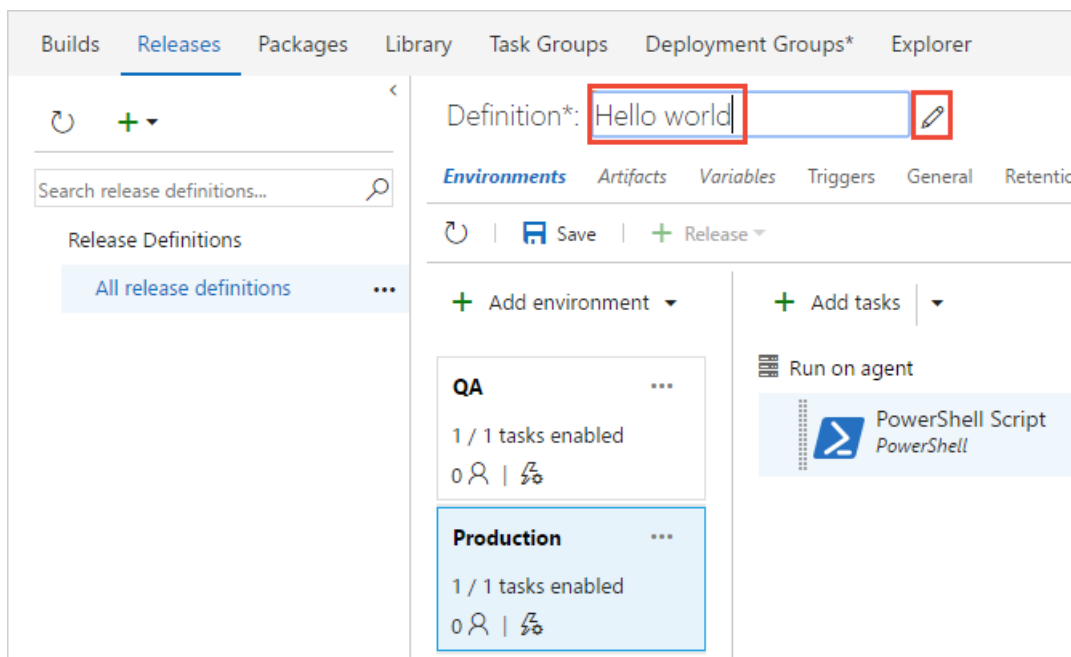
10. **Clone** the **QA** stage.



Leave **Automatically approve** and **Deploy automatically...** selected, and select **Create**.

11. Rename the new stage **Production**.

12. Rename the release pipeline **Hello world**.



13. 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.

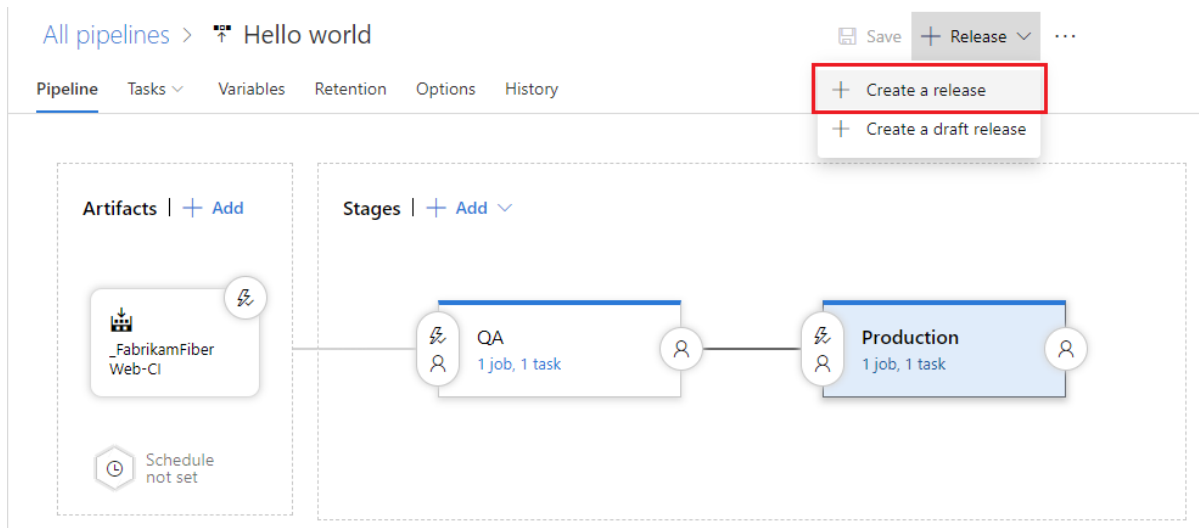
Also, notice that we used some variables in our script arguments. In this case, we used [release variables](#) instead of the build variables we used for the build pipeline.

Deploy a release

Run the script in each stage.

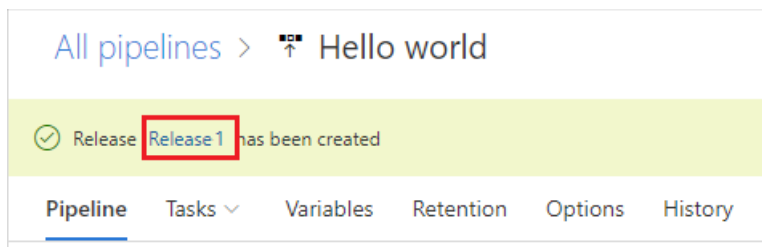
1. Create a new release.

- [New navigation](#)
- [Previous navigation](#)



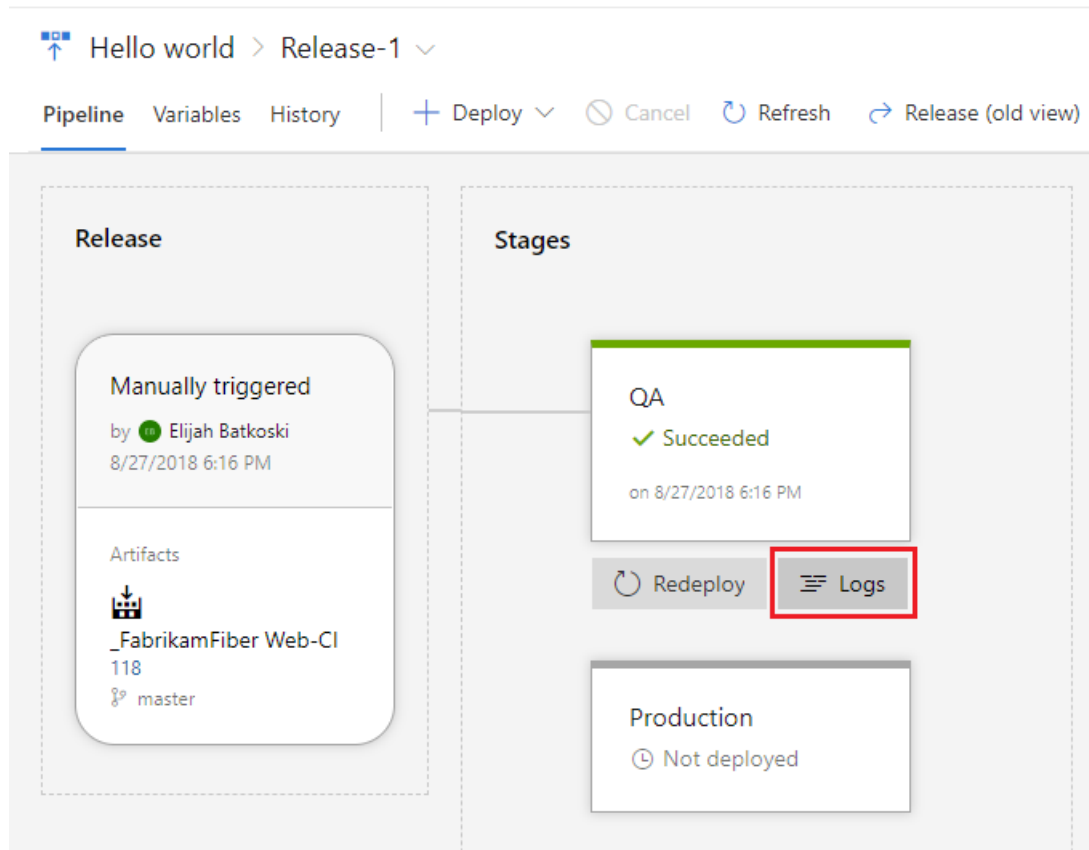
2. Define the trigger settings and artifact source for the release and then select **Create**.
3. Open the release that you just created.

- [New navigation](#)
- [Previous navigation](#)

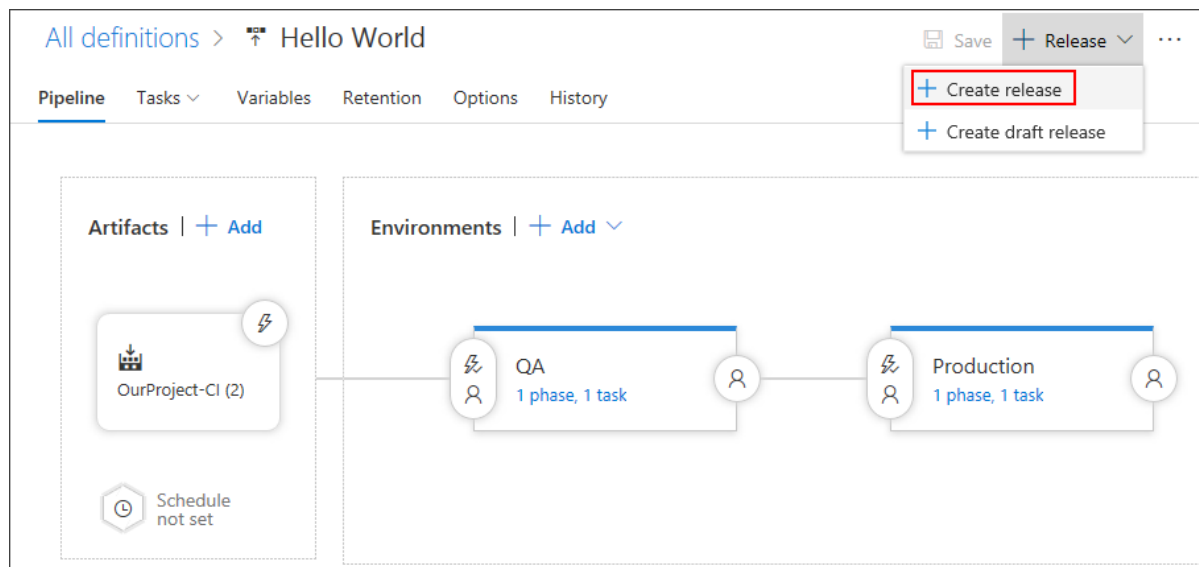


4. View the logs to get real-time data about the release.

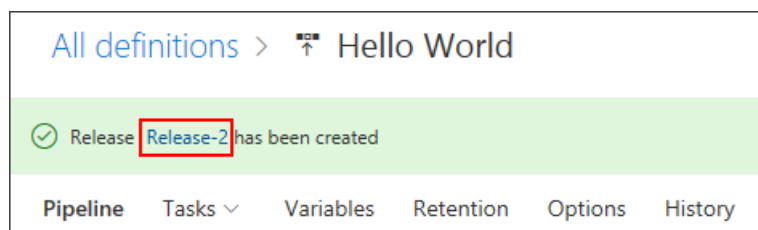
- [New navigation](#)
- [Previous navigation](#)



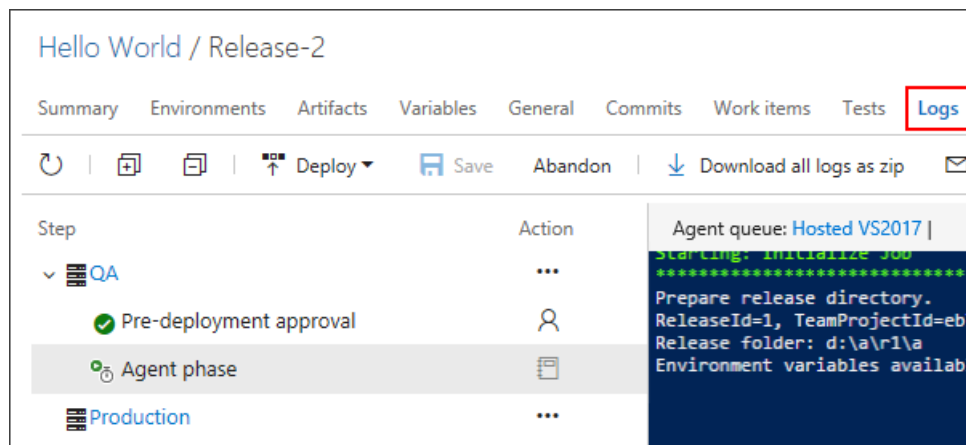
1. Create a new release.



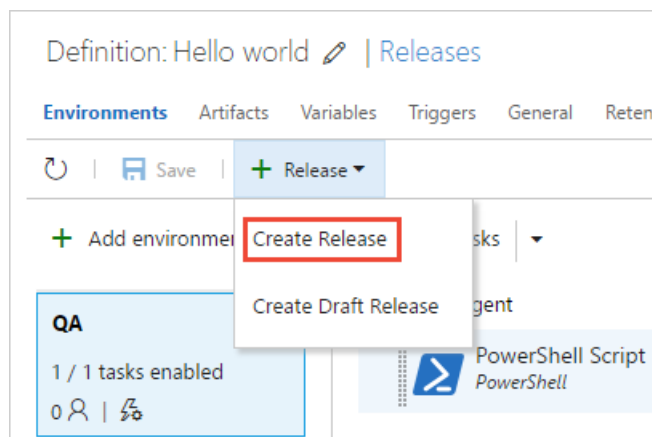
2. Define the trigger settings and artifact source for the release and then select **Queue**.
3. Open the release that you just created.



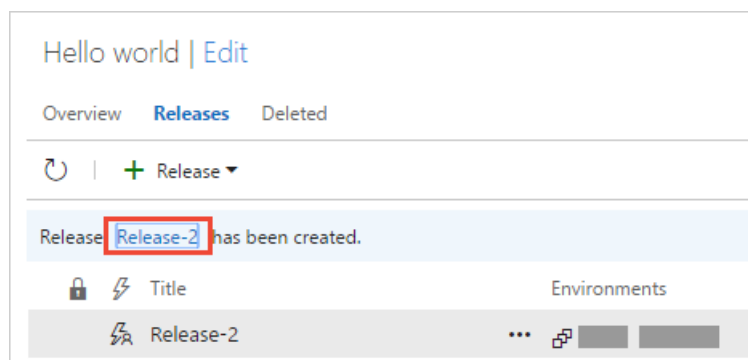
4. View the logs to get real-time data about the release.



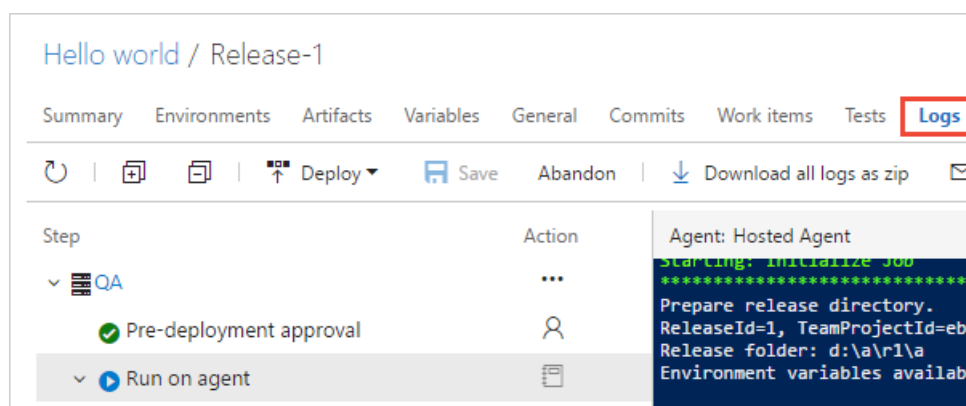
1. Create a new release.



2. Open the release that you just created.



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



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.

- [New navigation](#)
- [Previous navigation](#)

```
✓ PowerShell Script

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 -NonInteract
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
```

Hello World / Release-2

Summary Environments Artifacts Variables General Commits Work items Tests **Logs** History View All Details pane On

⌂ | 📁 | 📄 | Abandon | ⬇ Download all logs as zip | ✉ Send Email

Step	Action
> QA	...
▼ Production	...
✔ Pre-deployment approval	🔍
▼ ✔ Agent phase	📄
✔ Initialize Agent	📄
✔ Initialize Job	📄
✔ Download artifact - OurProject-CI (2)	📄
✔ PowerShell Script	📄
✔ Post-deployment approval	🔍

Agent queue: Hosted VS2017 | Agent: Hosted Agent
1 2018-04-25T14:53:54.9742188Z ##[section]Starting: PowerShell Scri
2 2018-04-25T14:53:54.9746441Z =====
3 2018-04-25T14:53:54.9746591Z Task : PowerShell
4 2018-04-25T14:53:54.9746697Z Description : Run a PowerShell scri
5 2018-04-25T14:53:54.9746796Z Version : 1.2.3
6 2018-04-25T14:53:54.9746897Z Author : Microsoft Corporation
7 2018-04-25T14:53:54.9747808Z Help : [More Information](ht
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
13 2018-04-25T14:53:57.3058801Z ##[section]Finishing: PowerShell Scr
14

Hello world / Release-7

Summary
Environments
Artifacts
Variables
General
Commits
Work items
Tests
Logs
History

Deploy
Save
Abandon
Download all logs as zip
Send Email

Step	Action	Agent: Hosted Agent
> QA	...	1 2017-04-11T12:54:58.5891186Z ##[section]Starting: PowerShell Script
Production	...	2 2017-04-11T12:54:58.6047463Z =====
Pre-deployment approval		3 2017-04-11T12:54:58.6047463Z Task : PowerShell
Run on agent		4 2017-04-11T12:54:58.6047463Z Description : Run a PowerShell script
Initialize Agent		5 2017-04-11T12:54:58.6047463Z Version : 1.2.3
Initialize Job		6 2017-04-11T12:54:58.6047463Z Author : Microsoft Corporation
Download Artifacts		7 2017-04-11T12:54:58.6047463Z Help : [More Information](https://go.micro
PowerShell Script		8 2017-04-11T12:54:58.6047463Z =====
Post-deployment approval		9 2017-04-11T12:54:58.6672458Z ##[command]. 'd:\a\r1\Hello world\drop\HelloWorld
		10 2017-04-11T12:54:59.3703946Z Hello world from Raisa Pokrovskaya
		11 2017-04-11T12:54:59.3703946Z Trigger: Hello world
		12 2017-04-11T12:54:59.3703946Z Now that you've got CI/CD, you can automatically d
		13 2017-04-11T12:54:59.4641245Z ##[section]Finishing: PowerShell Script
		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 the basics of using the visual designer to create and run 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
JavaScript	Node.js
Xcode	Xcode

Q & A

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

[What is Continuous Integration?](#)

[What is Continuous Delivery?](#)

[What is DevOps?](#)

What kinds of version control can I use

We've used a Git repository in Azure Repos to keep things focused on CI/CD for this tutorial.

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](#)
 - [Visual Studio with TFVC](#)
 - [Eclipse](#)
 - [Xcode](#)
 - [IntelliJ](#)
 - [Command line](#)
- Services
 - [Azure Pipelines](#)
 - Git service providers such as GitHub and Bitbucket
 - 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.

- [New navigation](#)
- [Previous navigation](#)

BuildsReleasesPackagesLibraryTask GroupsDeployment Groups*

Build Definitions

Search all definitions

+ New

MineAll DefinitionsQueuedXAML

↑ Folder / NameDefault branch summaryQueued

✓HelloWorld-CI

OurProject-CI

OurProject-Nightly

OurProject-PreDeployment

☆

...

Queue new build...

Move definition

View definition summary

Edit...

Add to my favorites

Add to team favorites

Clone...

Export

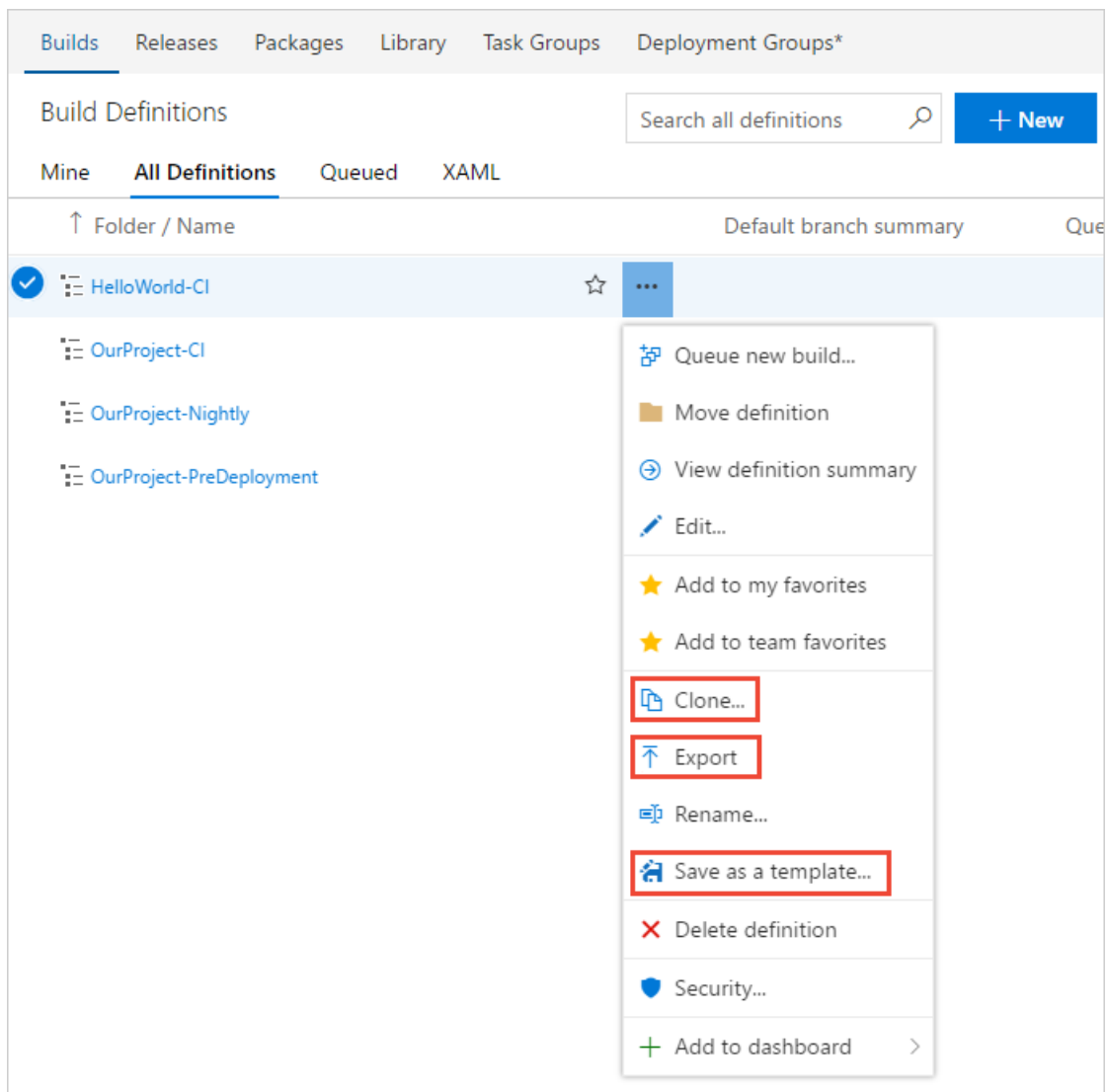
Rename...

Save as a template...

Delete definition

Security...

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.

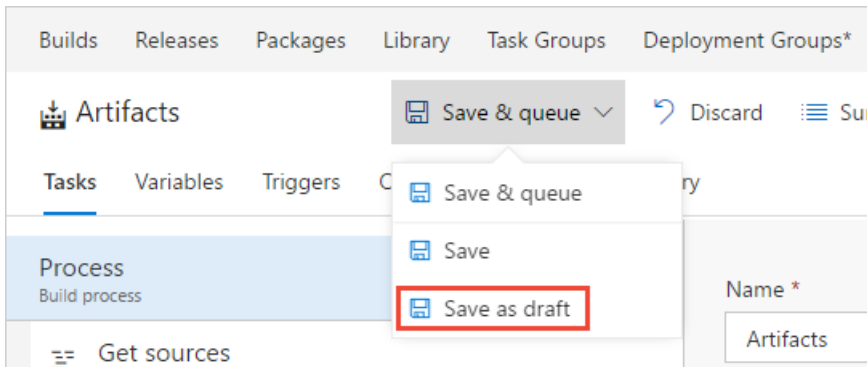
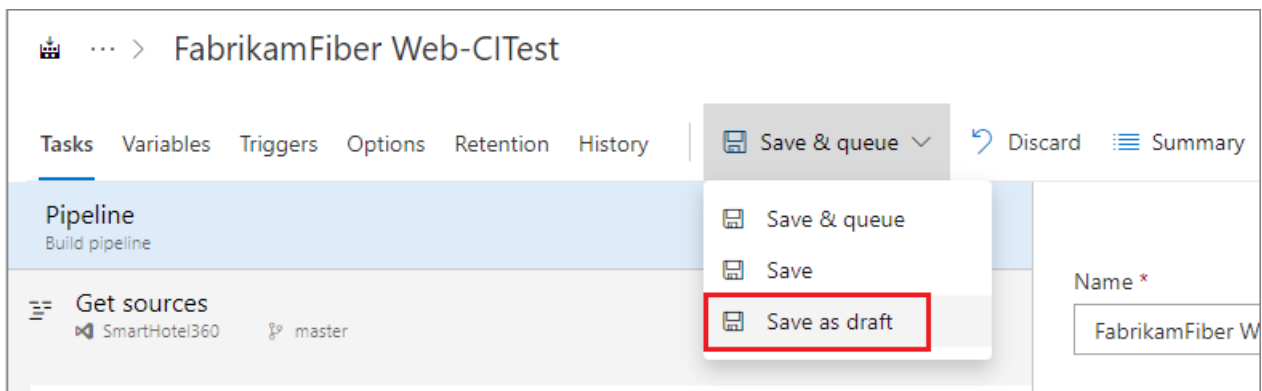
TIP

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?

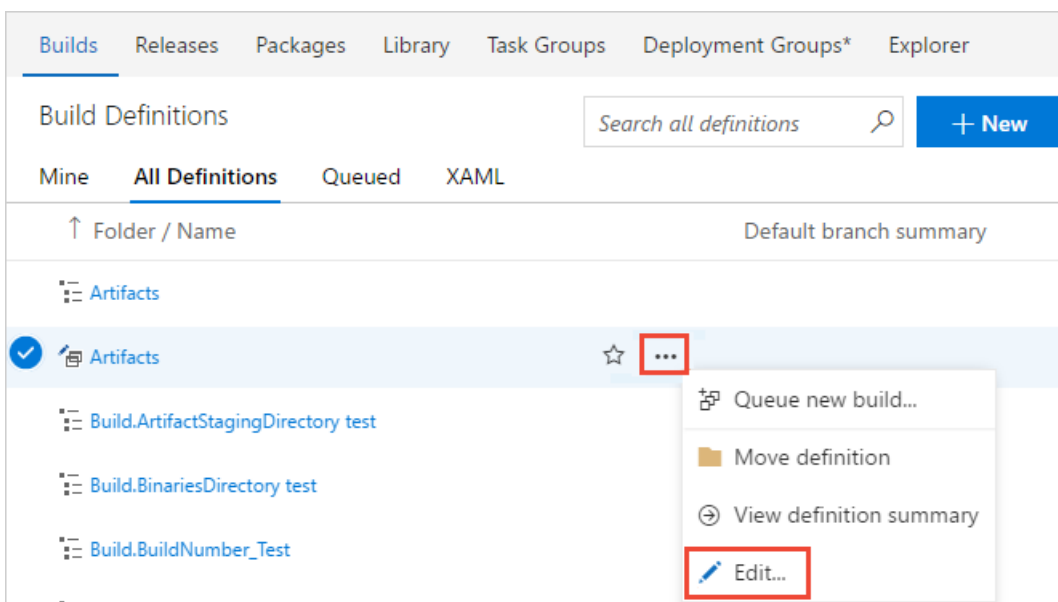
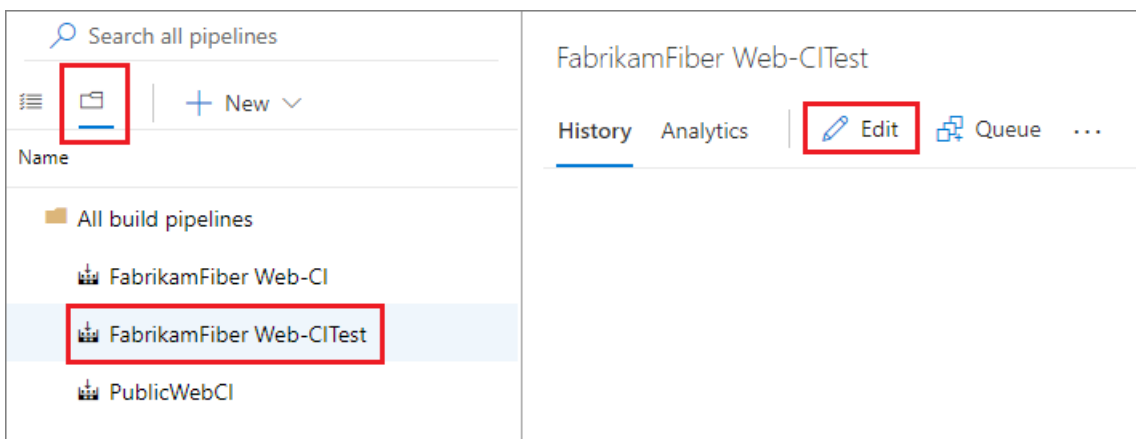
If you're editing a build pipeline and you want to test some changes that are not yet ready for production, you can save it as a draft.

- [New navigation](#)
- [Previous navigation](#)



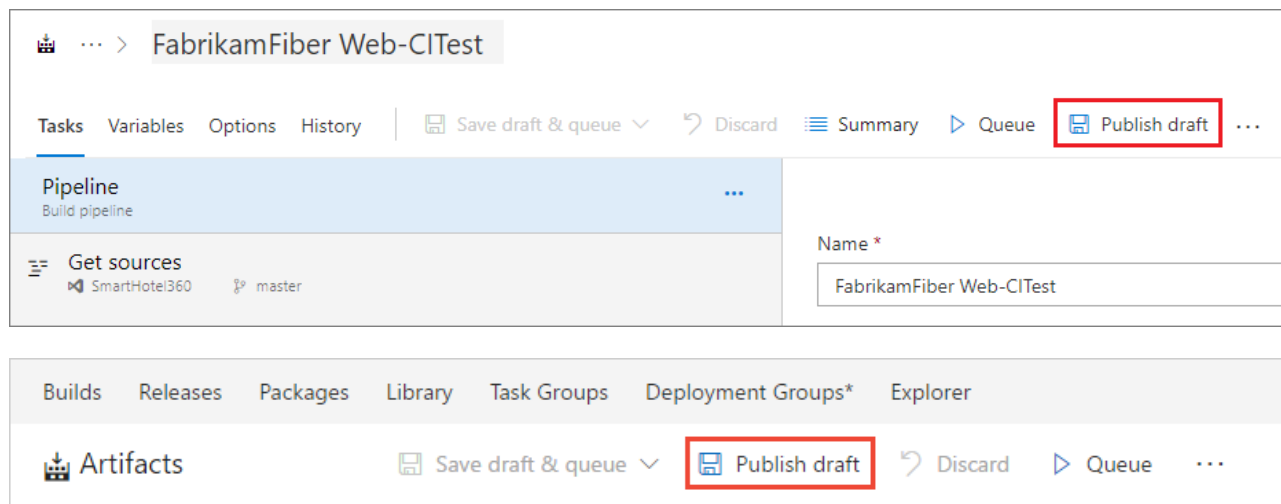
You can edit and test your draft as needed.

- [New navigation](#)
- [Previous navigation](#)



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

- [New navigation](#)
- [Previous navigation](#)



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

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](#).)

Where can I learn more about build pipeline settings?

To learn more about build pipeline settings, see:

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

9/10/2018 • 2 minutes to read • [Edit Online](#)

Azure Boards | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

You add work items to plan and manage your project. You use different types of work items to track different types of work—such as user stories or product backlog items, tasks, bugs, or issues. You can describe the work to be done, assign work, track status, and coordinate efforts within your team.

Here we show how to add work items from the web portal and view work items you've created.

Prerequisites

You can start adding work items once you connect to a team project. If you don't have a team project yet, create one in [Azure DevOps](#).

Open the Work Items page

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

NOTE

The **New navigation** feature, which provides a vertical navigation experience, is in preview for Azure DevOps. [Go here to enable it](#). When you enable **New navigation**, you automatically enable several new Agile tool features described in the [New Work Hubs](#) blog post. > For on-premises TFS users, choose **Previous navigation** for guidance.

- [New navigation](#)
- [Previous navigation](#)

(1) Check that you have selected the right project, then (2) choose **Boards>Work Items**.

NOTE

Depending on the process chosen when the project was created—[Agile](#), [Scrum](#), or [CMMI](#)—the types of work items you can create will differ. For example, backlog items may be called user stories (Agile), product backlog items (Scrum), or requirements (CMMI). All three are similar: they describe the customer value to deliver and the work to be performed.

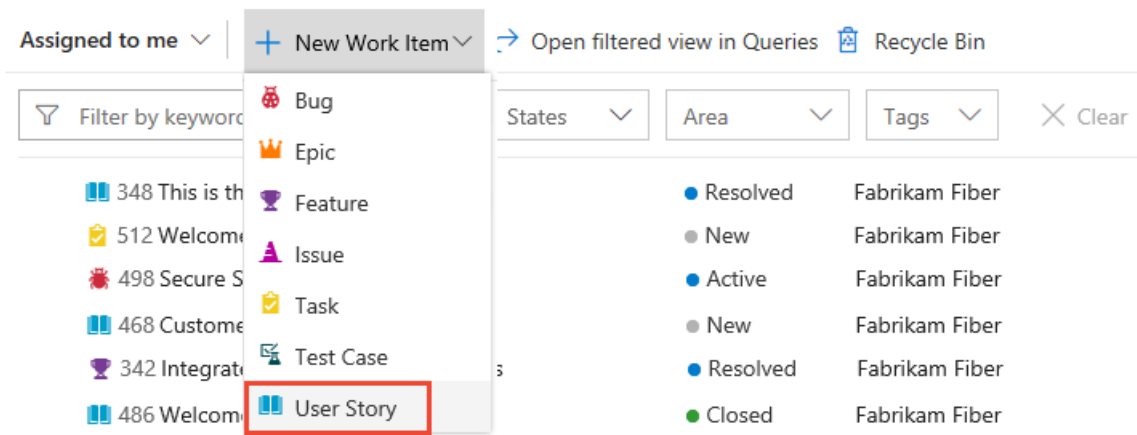
For an overview of all three processes, see [Choose a process](#).

Add a work item

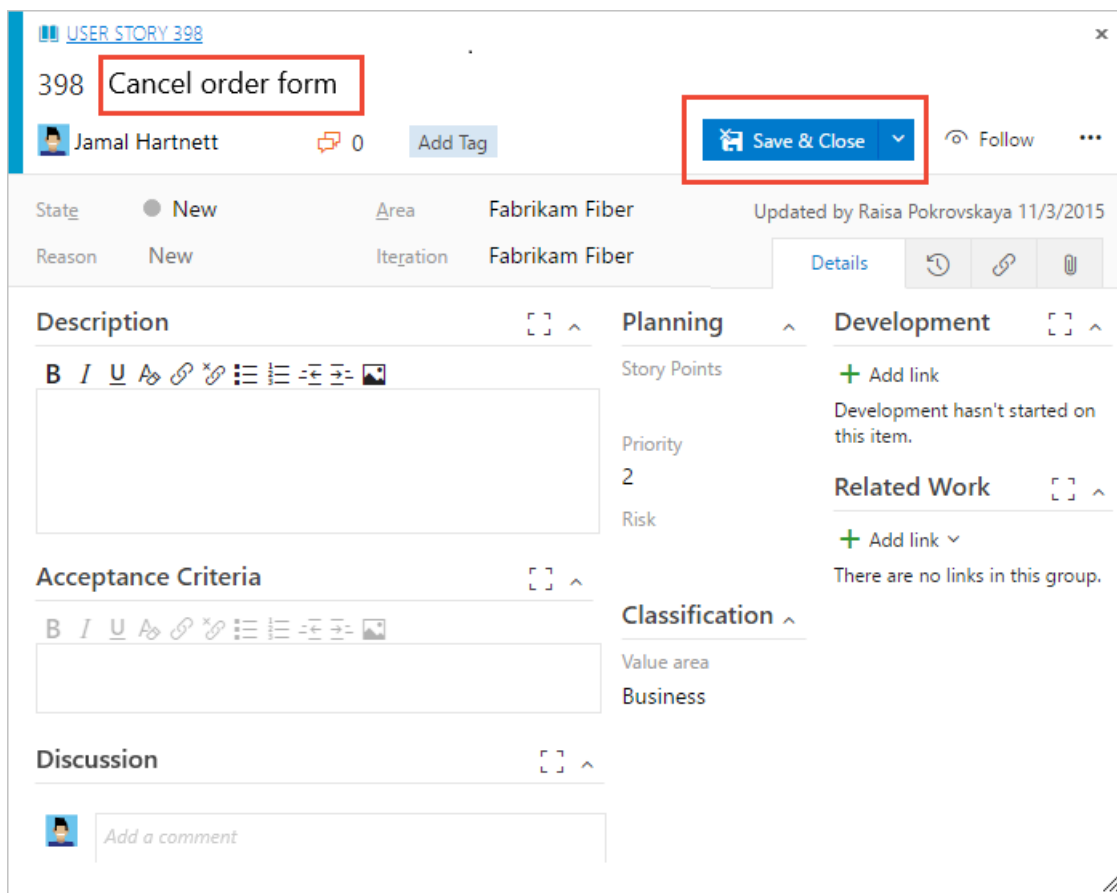
1. Adding a work item is just one click away. Simply choose the work item type from the **New Work Item** drop down menu.

For example, here we choose User Story.

Work Items



2. Enter a title and then save the work item. Before you can change the State from its initial default, you must save it.



You can [add tags to any work item](#) to filter backlogs, queries, and work item lists.

That's it!

Create as many work items as you need of the type you need to track the work you want to manage.

View the work items you've just created

Using the drop-down menu, 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.

Work Items

Assigned to me ▾

Assigned to me

Following

Mentioned

My activity

Recently updated

Recently completed

Recently created

+

- **Assigned to me:** lists all work items assigned to you in the project in the order they were last updated. To open or update a work item, simply click its title.
- **Following:** lists work items that you've elected to follow.
- **Mentioned:** lists work items in which you've been mentioned in the last 30 days.
- **My activity:** lists work items that you have recently viewed or updated.
- **Recently updated:** lists work items recently updated in the project.
- **Recently completed:** lists work items completed or closed in the project.
- **Recently created:** lists work items created within the last 30 days in the project.

For example, choose **My activity** to list all work items you've recently viewed, created, or modified.

Work Items

Assigned to me

Following

Mentioned

My activity

Recently created

+

 New Work Item ▾

529

 Cancel order form

Unassigned

New

To view any work item listed, choose the title.

For more information on using **Work Items**, see [View and add work items](#).

Try this next

[Create your backlog Kanban quickstart](#)

Or, [learn more about planning and tracking work](#).

Quickstart: Add, run, and update inline tests

9/11/2018 • 4 minutes to read • [Edit Online](#)

Azure DevOps Services

In this quickstart, you learn how to add, run, and update inline tests. A quick and easy way to start manual testing is to add the test to the user story or bug you want to test. From the Kanban board, you can quickly define inline tests, or a set of manual tests, for a backlog item. Not only can you add tests, you can run them and update their status. If you're new to working with the Kanban board, see [Kanban quickstart](#).

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

Open your Kanban board

NOTE


The **New navigation** feature provides a vertical navigation experience and is in preview for Azure DevOps Services. When you [enable new navigation](#), you automatically enable several new Agile tool features that are described in the [New Work Hubs](#) blog post.

On-premises Microsoft Team Foundation Server users can select **Previous navigation** for guidance.

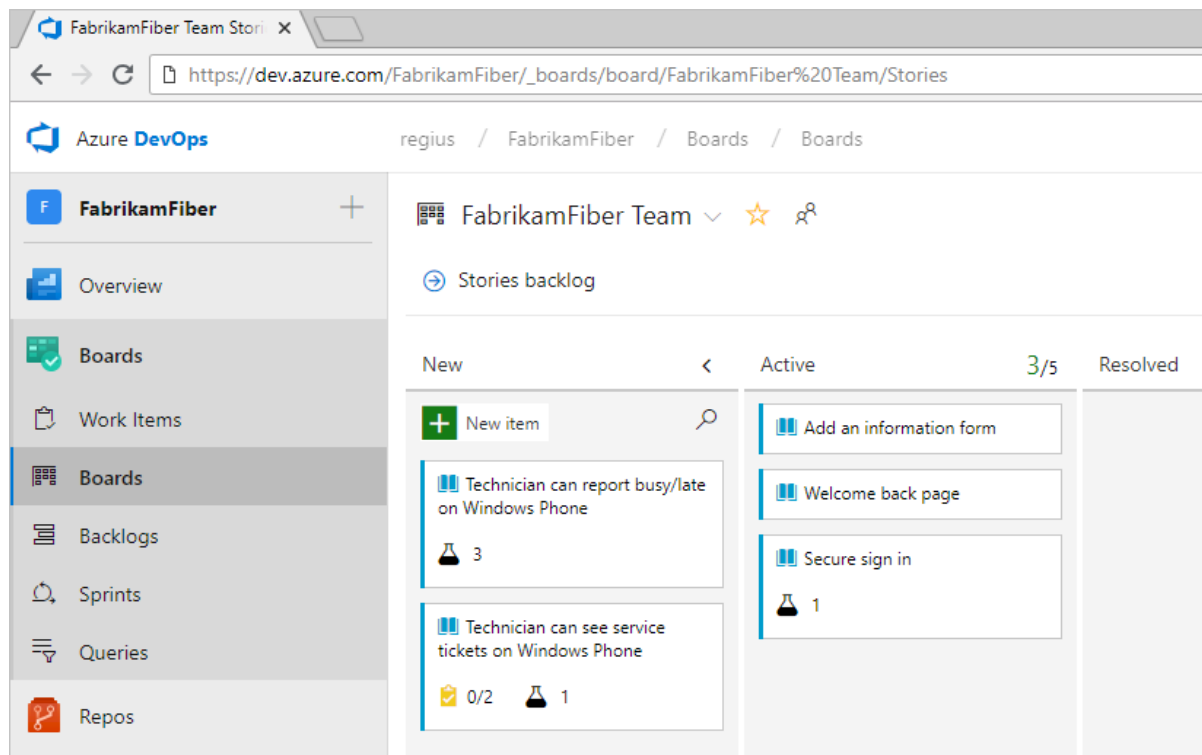
- [New navigation](#)
- [Previous navigation](#)

1. From your web browser, open the project for your Azure DevOps 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, choose the  Azure DevOps icon to [browse all projects and teams](#).

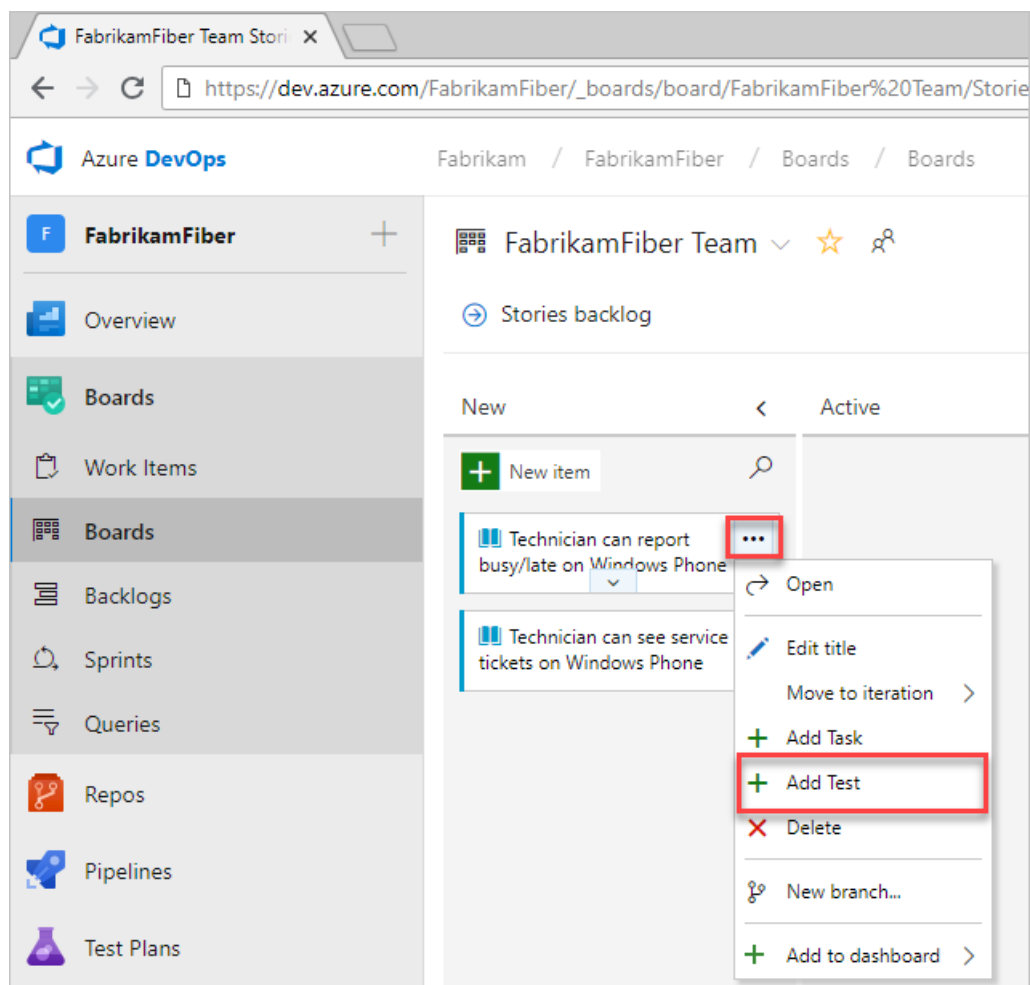
2. Select **Boards** to open the Kanban board.



Add tests

- [New navigation](#)
- [Previous navigation](#)

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



Adding inline tests is the same as adding test cases to a test suite. A default test plan and test suite are automatically 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, which has two manual tests defined with IDs of 337 and 341.

FabrikamFiber / FabrikamFiber / Test / Test Plans*

Test Plans > FabrikamFiber-tfvc Team_Stories_Sprint 3

FabrikamFiber-tfvc Team_Stories_Sprint 3

314 : Technician can see service tickets on...

Test suite: 314 : Technician can see service tickets on

Tests Charts

Run

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, simply keep entering each title and choose **Enter**.

New

New item

Technician can see service tickets on Windows Phone

2

+ Add Test | ↵

Change colors on initial view

Change initial page size

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 a web-based test case management interface. At the top, the test case is titled '337 Change colors on initial view' and is owned by 'Jamal Hartnett'. It has '0 comments' and a 'Save & Close' button. The test case is in the 'Design' state, located in the 'FabrikamFiber' area, specifically in the 'Release 1\Sprint 3' iteration. The 'Reason' is 'New'. Below the header, there are tabs for 'Steps', 'Summary', and 'Associated Automation'. The 'Steps' tab is active, showing a table with columns 'Steps' and 'Action'. A placeholder text 'Click or type here to add a step' is visible. To the right of the steps table, there are sections for 'Development' (with an 'Add link' button and a message 'Development hasn't started on this item.'), 'Related Work' (with an 'Add link' button and a list of related tests, including '314 Technician can see serv...'), and 'Details' (showing 'Priority 2' and 'Automation status Not Automated'). At the bottom, there is a 'Parameter values' section.

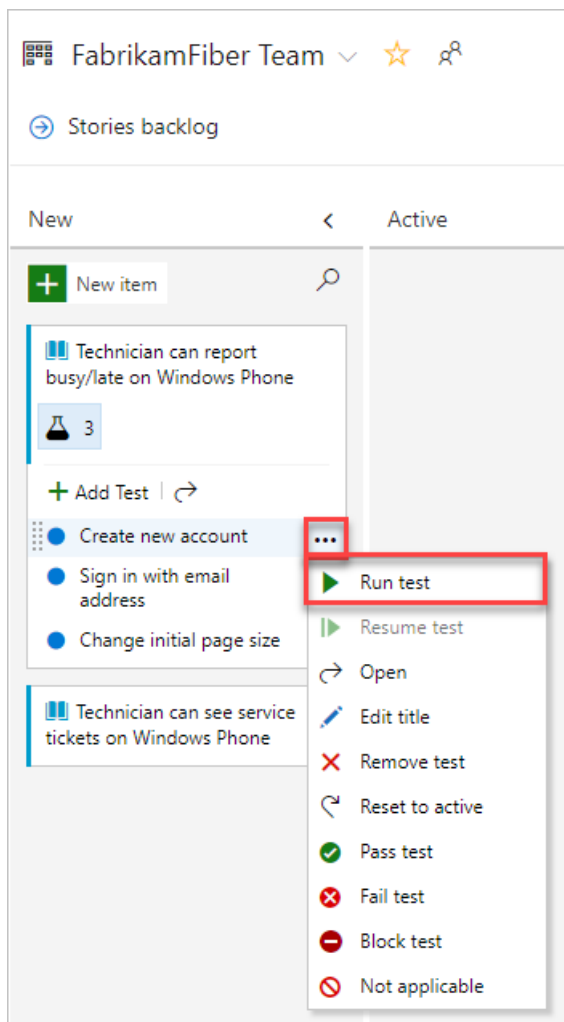
See [Create manual tests](#) to learn more about defining tests.

Prior to running the test, you must add details.

Run a test

- [New navigation](#)
- [Previous navigation](#)

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

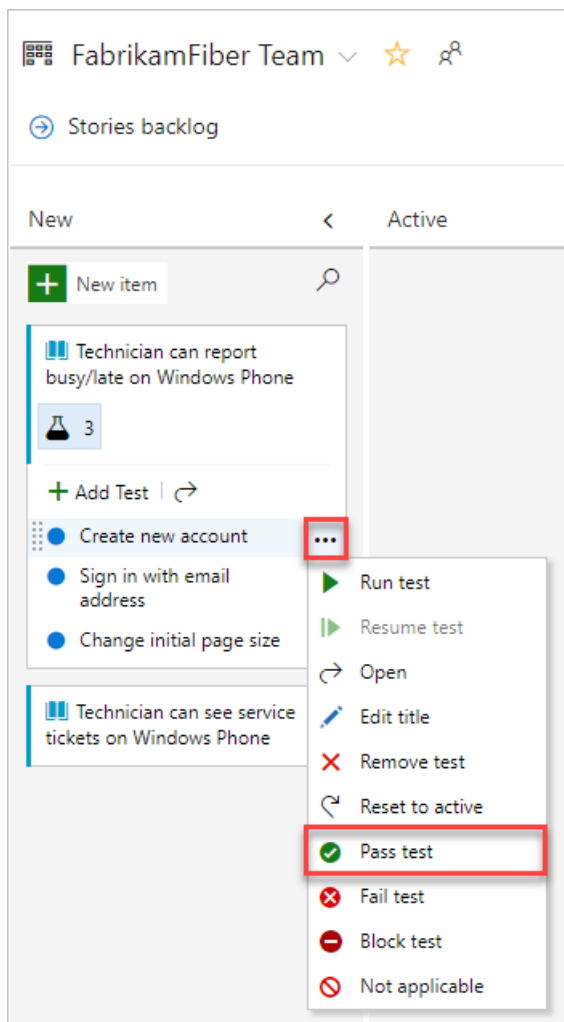


Microsoft Test Runner starts in a new browser instance. For details on running a test, see [Run manual tests](#).

Update the status of a test

- [New navigation](#)
- [Previous navigation](#)

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

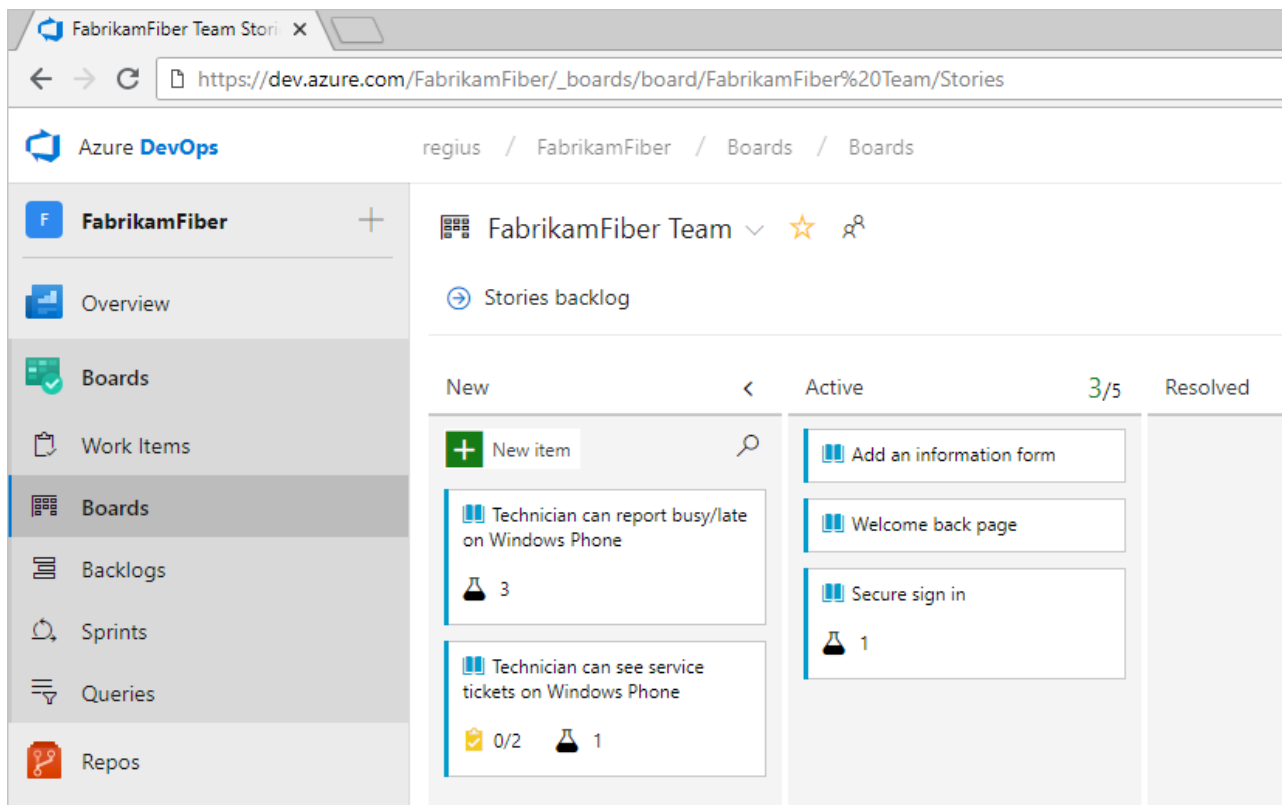


Updating the status of tests enable you to [track test results](#).

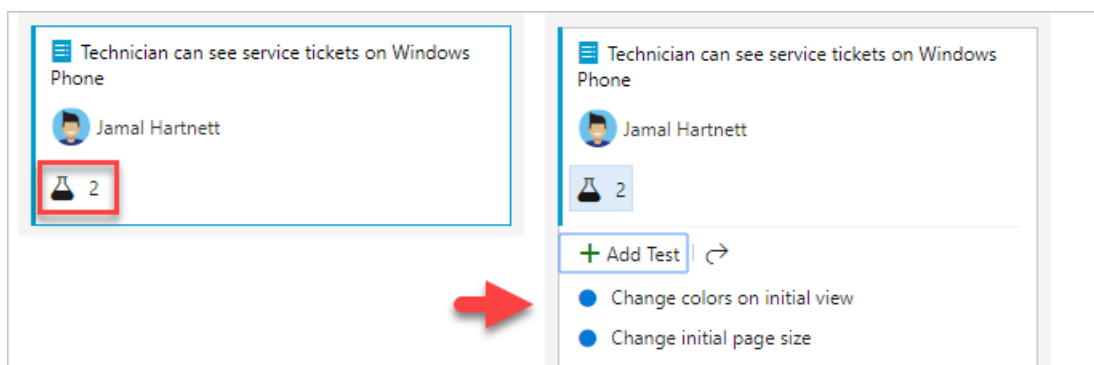
Expand or collapse inline tests

- [New navigation](#)
- [Previous navigation](#)

Upon first opening the Kanban board, you'll see an un-expanded view of checklists and tests.



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



Next steps

Use inline tests for lightweight traceability and to manage manual tests for user stories or other backlog items that they support.

[Learn more about test case management Kanban quickstart](#)

To initiate web-based exploratory testing for a user story, you need to install the Exploratory testing, see [Exploratory test your web app directly in your browser](#).

Quickstart: View permissions for yourself or others

9/10/2018 • 3 minutes to read • [Edit Online](#)

Azure DevOps Services | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

In this quickstart, you learn how to view your permissions or those defined for another which have been set in Azure DevOps Services or Team Foundation Server (TFS). That way, 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 [About permissions and groups](#). So to view the permissions you have, you need to open the permissions at the object, project, or collection level.

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.

NOTE

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

View project-level permissions

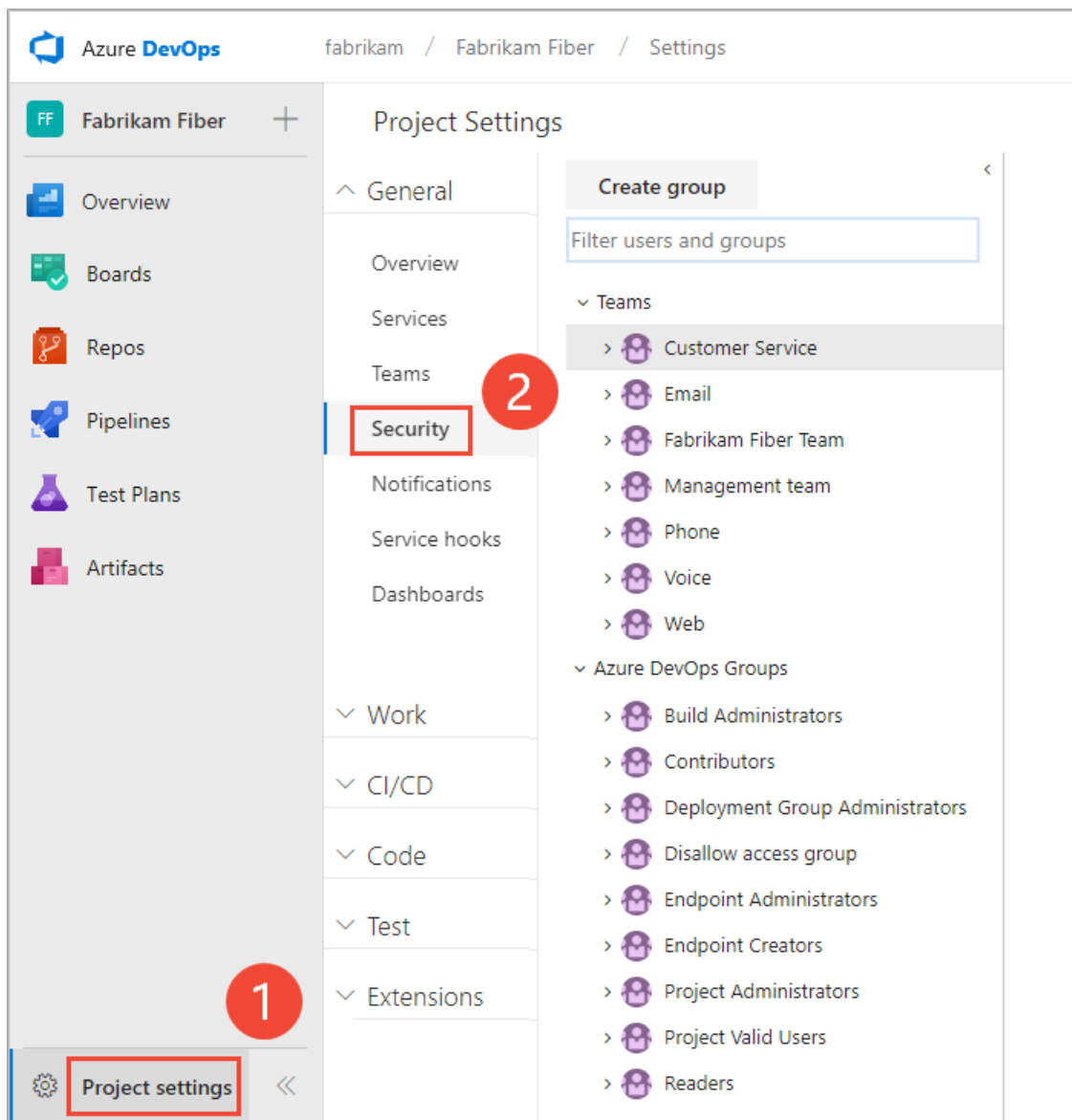
NOTE

Choose **Previous navigation** when you see a top-level blue bar. Choose **New navigation** if you see a vertical sidebar or if you enabled the **New Navigation** preview feature. The vertical sidebar, along with other navigational features, is enabled when the **New Navigation** preview feature has been enabled for the signed-in user or the organization. To learn how to use the web portal effectively, see [Web portal navigation](#).

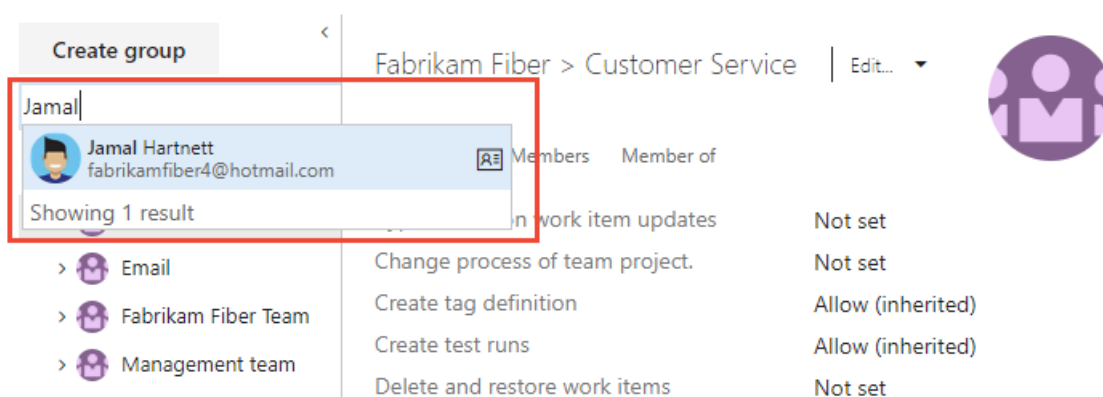
For on-premises TFS, choose **Previous Navigation** for guidance.

- [New navigation](#)
- [Previous navigation](#)

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



2. Begin typing the name into the *Filter users and groups* box. The system will automatically show the names that begin with the characters you type.



3. Choose the name you want. The project-level permissions you have set are based on the groups you belong to or those specifically set for your account.

Create group

Jamal

Jamal Hartnett

fabrikam > Jamal Hartnett | Edit...

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	Not set
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
Update project visibility	Not set
View analytics	Allow (inherited)
View project-level information	Allow (inherited)
View test runs	Allow (inherited)

Clear explicit permissions

Save changes

Undo changes

4. Choose **Member of** to see which security groups the user belongs to.

Here we see that *Jamal Hartnett* belongs to several teams as well as the Project Collection Administrators group.

Create group

Jamal

Jamal Hartnett

fabrikam > Jamal Hartnett | Edit...

Permissions

Member of

+ Add...

↺

Search


Display Name	Username Or Scope
Customer Service	[Fabrikam Fiber]
Fabrikam Fiber Team	[Fabrikam Fiber]
Web	[Fabrikam Fiber]
Project Collection Administrators	[fabrikam]
Project Administrators	[MyFirstProject]

New navigation isn't supported on TFS at this time. Choose **Previous navigation** for guidance.

View collection-level permissions

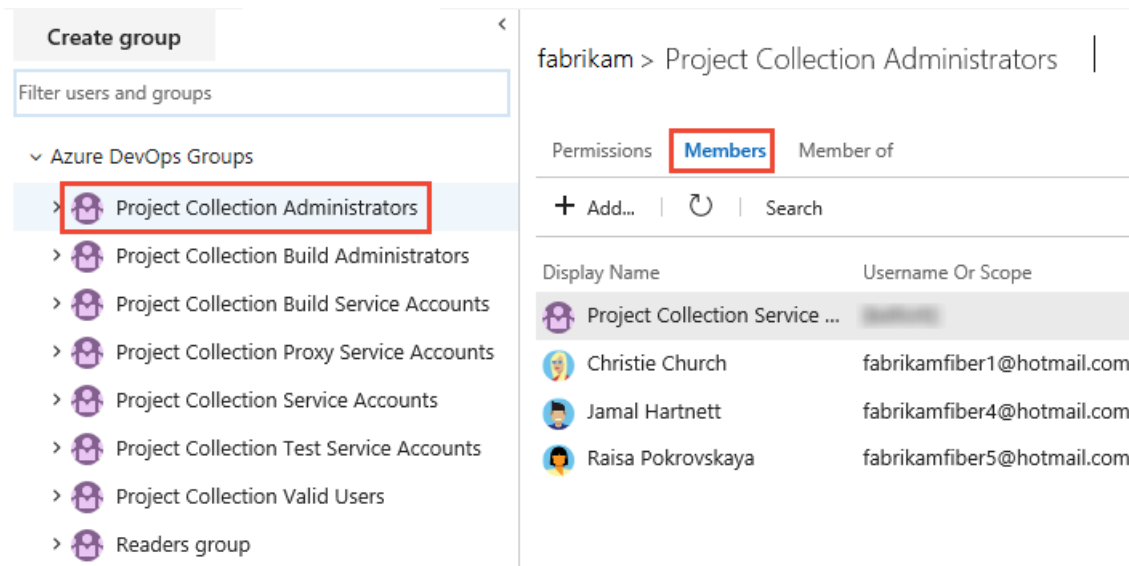
Open admin settings for the organization or a collection.

- [New navigation](#)
- [Previous navigation](#)

1. Choose the  Azure DevOps logo to open **Projects**. Then choose **Admin settings**.



2. Choose **Security**, the **Project Collection Administrators** group, and then **Members**.



3. Follow steps 2 through 4 in the procedure outlined previously for view project-level permissions.

New navigation isn't supported on TFS at this time. Choose **Previous navigation** for guidance.

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:

WIKI & DASHBOARD	DEVOPS	WORK TRACKING
<ul style="list-style-type: none">- README & Wiki- Dashboards	<ul style="list-style-type: none">- Git branch- Git repository- TFVC- Builds- Release pipeline security- Approvals and approvers	<ul style="list-style-type: none">- Area and iteration paths- Work item query and folder- Plans

Next steps

[Look up the organization owner or a project administrator](#)

Quickstart: Sign up for a free Azure DevOps organization

9/10/2018 • 3 minutes to read • [Edit Online](#)

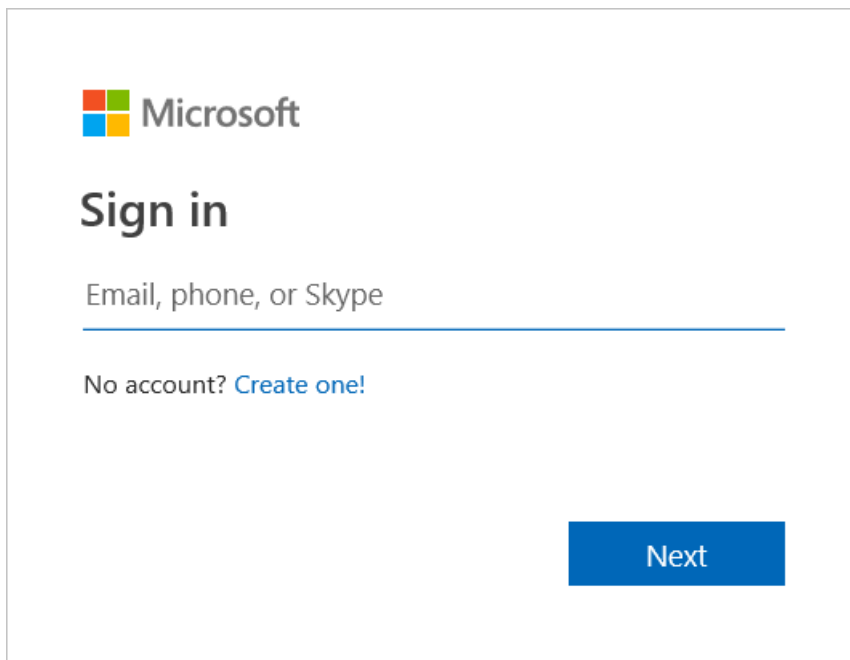
In this quickstart, you learn how to sign up for a free Azure DevOps organization with a Microsoft account and how to invite your teammates.

Sign up for Azure DevOps Services to upload and share code in free, unlimited private Git repositories.

Then, connect to your favorite development tool like Eclipse, Xcode, Visual Studio, IntelliJ, or Android Studio to work on apps anytime, anywhere.

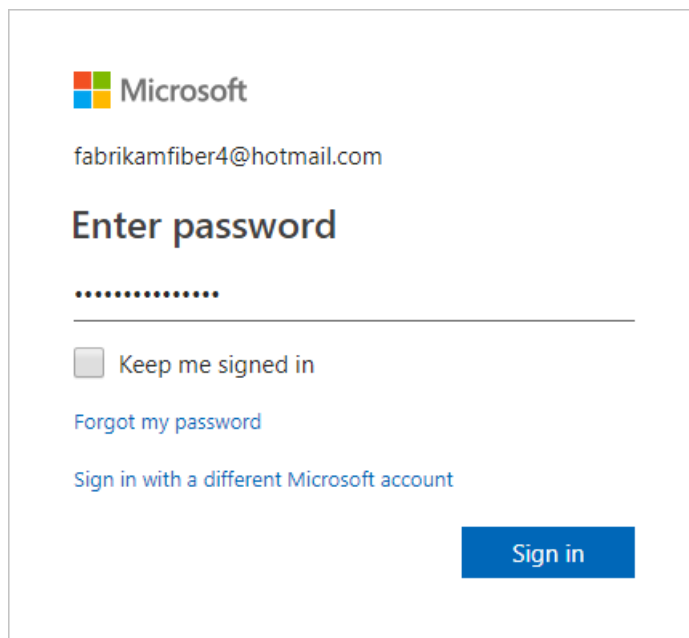
Sign up for Azure DevOps with a personal Microsoft account

1. Select the sign-up link for [Azure DevOps](#).
2. Enter your email address, phone number, or Skype ID for your Microsoft account. If you're a Visual Studio subscriber and get Azure DevOps as a benefit, use the Microsoft account associated with your subscription. Choose **Next**.

A screenshot of the Microsoft sign-in interface. At the top left is the Microsoft logo. Below it, the text "Sign in" is displayed in a large, bold font. Underneath "Sign in" is a text input field with the placeholder text "Email, phone, or Skype". Below the input field is a horizontal line. Under the line, the text "No account? [Create one!](#)" is visible. At the bottom right of the form is a blue button with the text "Next" in white.

3. Enter your password and select **Sign in**.

If you don't have a Microsoft account, you can [create a Microsoft account](#) at this time.



Microsoft

fabrikamfiber4@hotmail.com

Enter password

.....

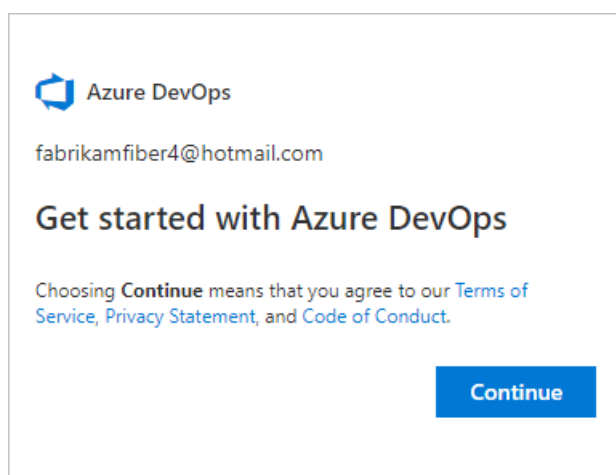
☐ Keep me signed in

[Forgot my password](#)

[Sign in with a different Microsoft account](#)

Sign in

4. To get started with Azure DevOps , choose **Continue**.



Azure DevOps

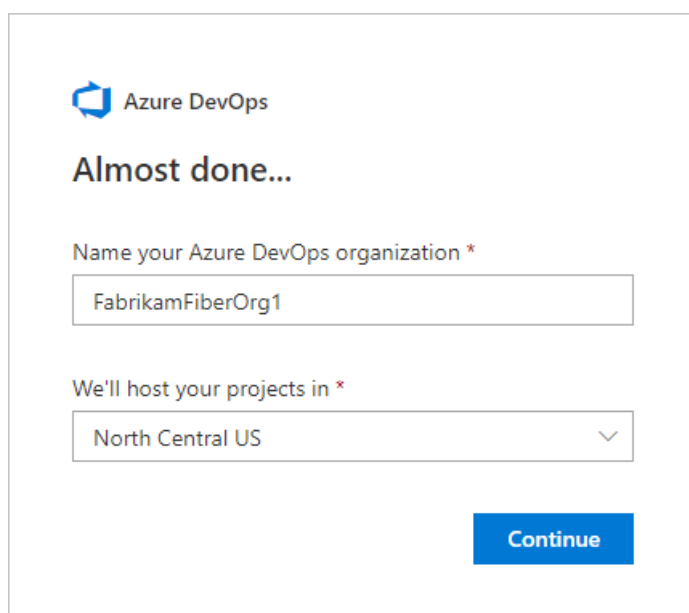
fabrikamfiber4@hotmail.com

Get started with Azure DevOps

Choosing **Continue** means that you agree to our [Terms of Service](#), [Privacy Statement](#), and [Code of Conduct](#).

Continue

5. Enter a name for your organization. The name you enter cannot contain spaces or special characters (such as / \ [] : | < > + = ; ? or *), cannot end in a period or comma, must be less than 256 characters, and must be unique within the DevOps namespace. You can also choose between several locations for where you want your data hosted. Select **Continue**.



Azure DevOps

Almost done...

Name your Azure DevOps organization *

FabrikamFiberOrg1

We'll host your projects in *

North Central US

Continue

You see the following dialog box as your organization is created.

Congratulations, you're now an organization owner!

To sign in to your organization at any time, go to `https://dev.azure.com/{yourorganization}`.

6. Enter a name for your project and select the visibility. The name you enter cannot contain spaces or special characters (such as `/ : \ ~ & % ; @ ' " ? < > | # $ * } { , + = []`), cannot begin with an underscore, cannot begin or end with a period, and must be 64 characters or less. Visibility can be either public or private. With public visibility, anyone on the internet can view your project. With private visibility, only people who you give access to can view your project. Select **Create project**.

Welcome to your project

When your project has been created, the welcome page appears.

NOTE

Your first project was created by using a Git repository and the [Agile process](#). If you want a project that uses the Team Foundation Version Control (TFVC) repository or the Scrum or CMMI process, see [Choose a process](#) for a comparison of processes. Then, you can choose a process by [adding another project](#).

Select one of the following tasks to get started:

- **Boards** to begin [adding work items](#).
- **Repos** to open the [Repos > Files](#) page. There, you can clone or import a repository or [initialize a README file](#) for your project summary page.
- **Pipelines** to start [defining a pipeline](#).
- **Test Plans** to start [defining test plans and test suites](#).
- [Manage your services](#) to disable the visibility of one or more services.

To get started managing your project, see [Get started as an administrator](#).

For more information about organizations and projects, see these articles:

- [Define organizations and projects](#)
- [About projects and scaling your organization](#)
- [Manage projects](#)

Invite team members

Give a team member access to your organization by adding their email address to your organization.

1. Sign in to your Azure DevOps organization (`https://dev.azure.com/{yourorganization}`).

2. Select  **Organization settings**.

3. Select **Users > Add new users**.

4. Complete the form by entering or selecting the following information:

- **Users:** Enter the email addresses (Microsoft account) for the users. You can add several email addresses by separating them with a semicolon (;). An email address appears in red when it's accepted.
- **Access level:** Leave the access level as **Basic** for users who will contribute to the code base. To learn more, see [About access levels](#).
- **Add to project:** Select the project you named in the preceding procedure.
- **DevOps Groups:** Leave as **Project Contributors**, the default security group for users who will contribute to your project. To learn more, see [Default permissions and access assignments](#).

NOTE

You must add email addresses for [personal Microsoft accounts](#) unless you plan to use [Azure Active Directory \(Azure AD\)](#) to authenticate users and control organization access. If a user doesn't have a Microsoft account, ask the user to [sign up](#) for a Microsoft account.

5. When you're done, select **Add** to complete your invitation.

For more information about managing users and organization access, see [Add organization users for Azure DevOps](#).

Next steps

[Add code to your Git repository](#)

[Plan and track work](#)

Quickstart: Create your Azure DevOps organization

9/11/2018 • 4 minutes to read • [Edit Online](#)

Azure DevOps Services

To create an Azure DevOps organization with a personal Microsoft account or a work or school account, go to the [Azure DevOps Services](#) site.

In this quickstart, you learn how to upload and share code in free, unlimited private Git repositories or Team Foundation Version Control. To work on apps anytime, anywhere, you can connect your favorite development tools, such as Eclipse, Xcode, Visual Studio, IntelliJ, or Android Studio. Azure DevOps Services offers integrated, enterprise agile tools for DevOps, so your team can build often, test early, and ship faster.

Want to set up an on-premises server? [Get Team Foundation Server](#), or learn [how to install and set up Team Foundation Server](#).

[Who can join for free? What do users get in Azure DevOps Services?](#)

Prerequisites

- Read [Defining your Azure DevOps organizations and projects](#).
- Before you begin, do either of the following:
 - Complete the following steps to use only Microsoft accounts with your organization. Ignore the Azure Active Directory (Azure AD) callouts.
 1. If you don't have a Microsoft account, create one when you sign up for Azure DevOps.
 2. 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.
 - Complete the following steps to authenticate users and control organization access through your Azure AD. Pay attention to the Azure AD callouts.
 1. Use your work or school account 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 capabilities](#).
 2. You need a work or school account that's managed by your Azure AD. If you use Azure or Office 365, you might have one already. If you don't, learn how to [sign up for Azure as an organization](#).
 3. To use existing on-premises identities with Azure DevOps Services, see [use Azure AD Connect for integrating on-premises directories with Azure AD](#).

How does Azure AD control access to Azure DevOps Services?

Azure DevOps Services authenticates users through your Azure AD, so that only users who are members in that directory can access your organization. When you remove users from that directory, they can no longer access your organization. Only specific [Azure AD administrators](#) can manage users in your directory, so administrators control who can access your organization.

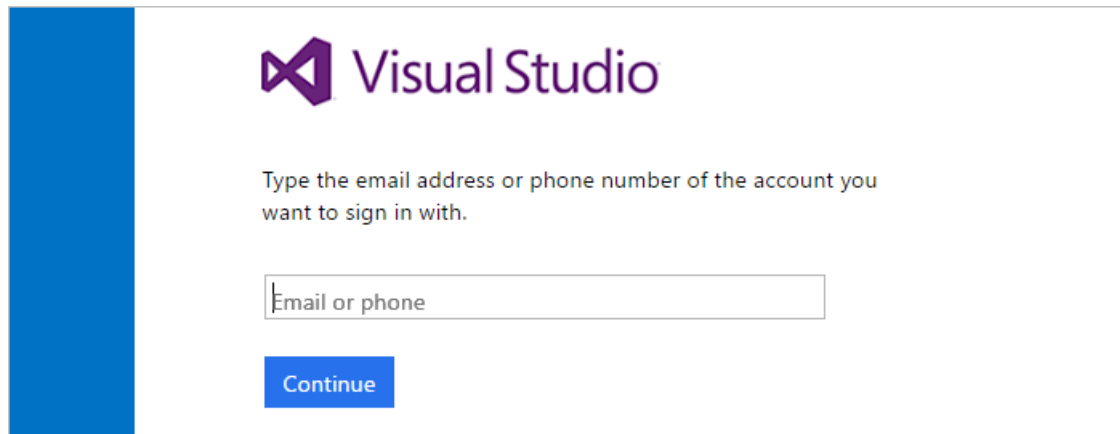
Without Azure AD, you're solely responsible for controlling Azure DevOps organization access. And all users must

sign in with their Microsoft account. [What are other differences?](#)

Create your organization and sign up for Azure DevOps Services

1. Go to [Azure DevOps Services](#), and use either of the following:

- **Microsoft account:** If you're a Visual Studio subscriber and get Azure DevOps Services as a benefit, use the Microsoft account email address that's associated with your subscription.
- **Azure AD:** Use your work or school account email address. Depending on the account you use, your sign-in page might vary from the page shown as follows:

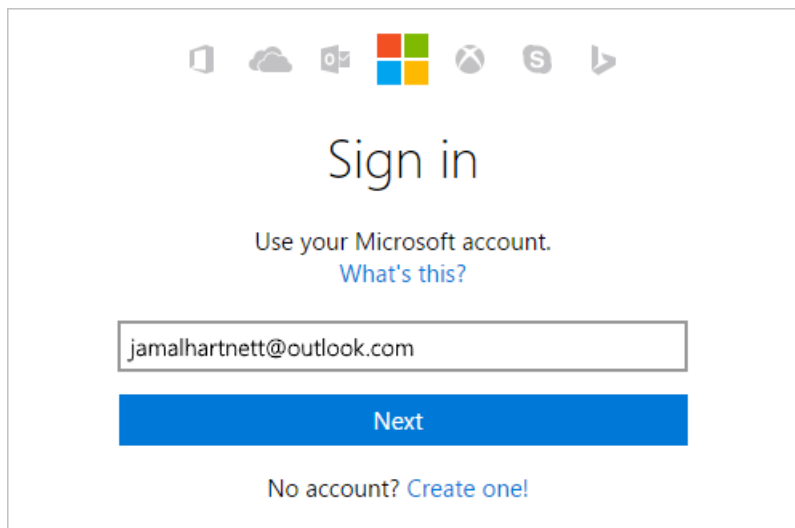
A screenshot of the Visual Studio sign-in page. It features a blue vertical bar on the left. The main area has the Visual Studio logo at the top. Below the logo, it says "Type the email address or phone number of the account you want to sign in with." There is a text input field with the placeholder "Email or phone" and a blue "Continue" button below it.

[Got browser problems?](#)

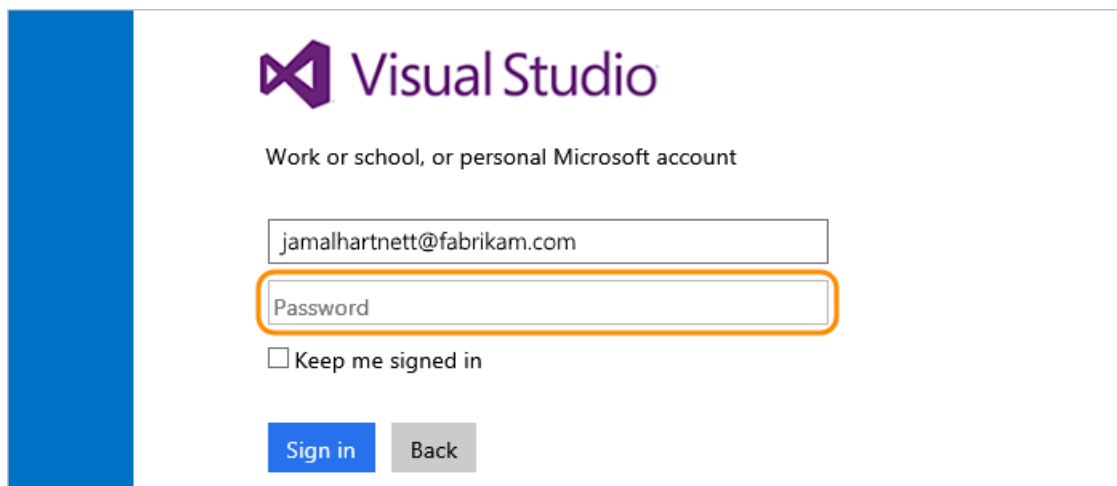
2. Do the following:

- a. **Microsoft account:** Enter the email address for your Microsoft account, select **Next**, and then enter your password to finish signing in.

If you are not using **Azure AD**, and you don't have a Microsoft account, you can create a Microsoft account at this time.

A screenshot of the Microsoft account sign-in page. At the top, there are several small icons representing different services. The main heading is "Sign in". Below it, it says "Use your Microsoft account." and "What's this?". There is a text input field containing the email address "jamalhartnett@outlook.com". Below the input field is a blue "Next" button. At the bottom, it says "No account? [Create one!](#)".

- b. **Azure AD:** On the Visual Studio sign-in page, enter your password for your work or school account, and then select **Sign in**.



Visual Studio

Work or school, or personal Microsoft account

jamalhartnett@fabrikam.com

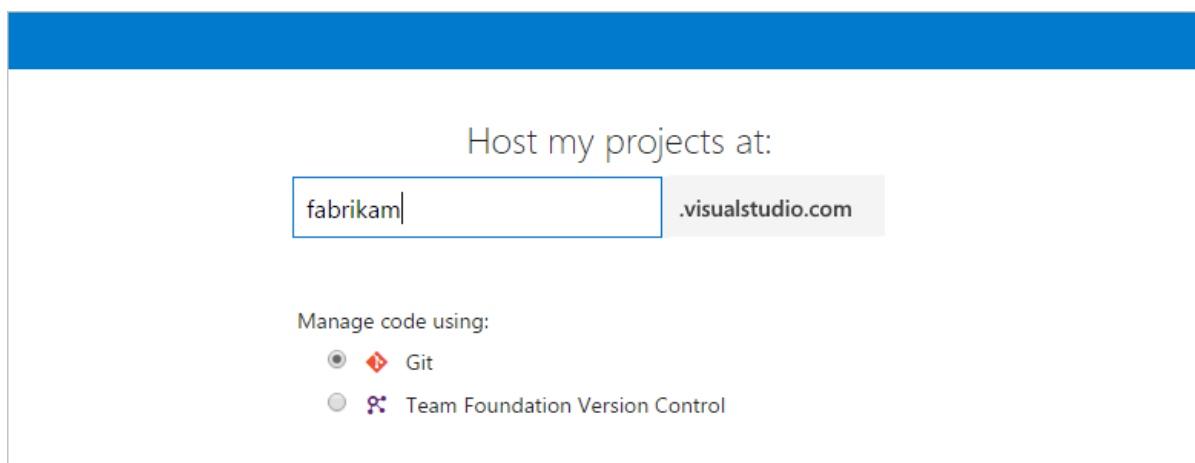
Password

☐ Keep me signed in

Sign in Back

Why am I asked to choose between my work or school account and my personal account?

- Under **Host my projects at**, enter the name of your organization and then, under **Manage code using**, select **Git** or **Team Foundation Version Control**.



Host my projects at:

fabrikam .visualstudio.com

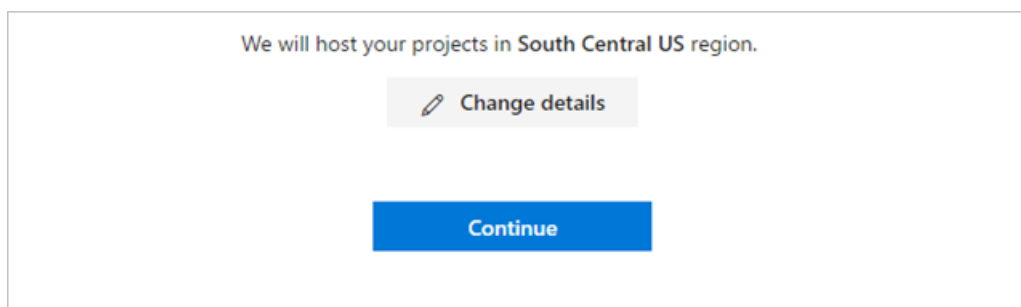
Manage code using:

☒ Git

☐ Team Foundation Version Control

Learn which version control works best for you: [Git](#) or [Team Foundation Version Control](#).

- Confirm your organization's location and, if you're using **Azure AD**, confirm the directory that you're connecting to your organization.

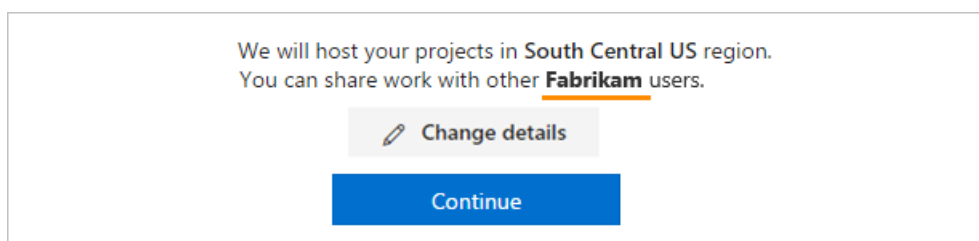


We will host your projects in **South Central US** region.

Change details

Continue

Azure AD:



We will host your projects in **South Central US** region.
You can share work with other **Fabrikam** users.

Change details

Continue

Microsoft account and Azure AD:

Azure DevOps Services creates your first project as *MyFirstProject* and uses Agile as your default work item process to organize your work.

Select **Change details** to [rename your project](#), [change the organization location](#), or [select another process](#), such as [Scrum](#).

Azure AD:

After you create your account, only members of the directory shown here can access your organization, or you must use [Azure AD business-to-business \(B2B\) collaboration capabilities](#) to add users from other organizations.

If you belong to multiple directories, ensure that you want to connect this directory to your organization. Changing the directory now is easier than [changing the directory later](#).

5. After your organization and project are created, add code, work items, or more users.

The screenshot shows the 'MyFirstProject' dashboard. At the top left is a pink square logo with a white 'M'. To its right is the text 'MyFirstProject' with a yellow star icon and the subtitle 'Briefly describe your project...'. Below this is an 'Add tags' button. The main heading is 'Get started with your new project!'. Underneath is a section for cloning the repository, with a dropdown arrow and the text 'Clone to your computer'. Below this is a code block showing the HTTPS URL: 'https://visualstudio.com/MyFirstProject/_git/MyFirstProject'. There are buttons for 'Generate Git credentials' and 'Clone in Visual Studio'. A note mentions having problems authenticating in Git. At the bottom, there are four options with dropdown arrows: 'or push an existing repository from command line', 'or import a repository', 'or initialize with a README or gitignore', and 'or build code from an external repository'. A 'Setup Build' button is at the bottom left. On the right side, there is a sidebar with sections: 'Members (1)' with a user icon and a plus sign, 'Activity', 'Code' (No code yet), 'Build & Release' (No builds yet), and 'Work' (No work items yet).

Congratulations, you're now an organization owner!

To sign in to your organization at any time, go to `https://<yourorganization>.visualstudio.com`.

NOTE

If you activated your Visual Studio subscription with a Microsoft account, and your subscription includes Azure DevOps Services as a benefit, learn [how to add your work or school account](#) to your subscription so that you can use your subscriber benefits in Azure DevOps Services.

Next steps

[Manage users and access](#) or

[Create project](#)

or

Add code to Git or Team Foundation Version Control:

- **Git:** Use [Eclipse](#), [Xcode](#), [Android Studio](#), [IntelliJ](#), [Visual Studio](#), or [Visual Studio Code](#).
- **Team Foundation Version Control:** Use [Eclipse](#), [Xcode](#), [Visual Studio](#), or [Visual Studio Code](#).

Related articles

- [Create your backlog](#)
- [Manage your process](#)
- [Customize your process](#)

Quickstart: Add users to a project or specific team

9/10/2018 • 8 minutes to read • [Edit Online](#)

Azure DevOps Services | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

In this quickstart, you learn how to add users to a project or specific team. For anyone to access a project, they must be added to one of the default security groups or a custom group. Usually you add them to the Contributors group. For a quick look at what permissions are assigned to the default groups, see [Permissions and access](#).

The easiest way to add a number of users to a project is to add groups defined in [Azure Active Directory \(Azure AD\)](#) or [Active Directory \(AD\)](#).

IMPORTANT

If you're adding users to an Azure DevOps organization and you don't use Azure AD, then you need to [add their "personal" Microsoft accounts to your account or project](#). After you've added them to one project, you can add them to additional projects using the procedures provided in this topic.

Once users have been added to a project, you can browse for that name by display name as well as account name. Also, you can [add them to a specific team](#). To add a team, see [Add a team](#).

Prerequisites

- You must have a project. If you don't have a project yet, [create one](#).
- To add users to a project, you must be a member of the [Project Administrators group](#) or have your **Edit project-level information** set to **Allow**. Note that you can add Stakeholders to the Project Administrators group and then they can add users to an organization or project.
- To add users to a team, you must have been [added as a team administrator for the team](#), or you must be a member of the Project Administrators Group or have your **Edit project-level information** set to **Allow**.

Add users to a project

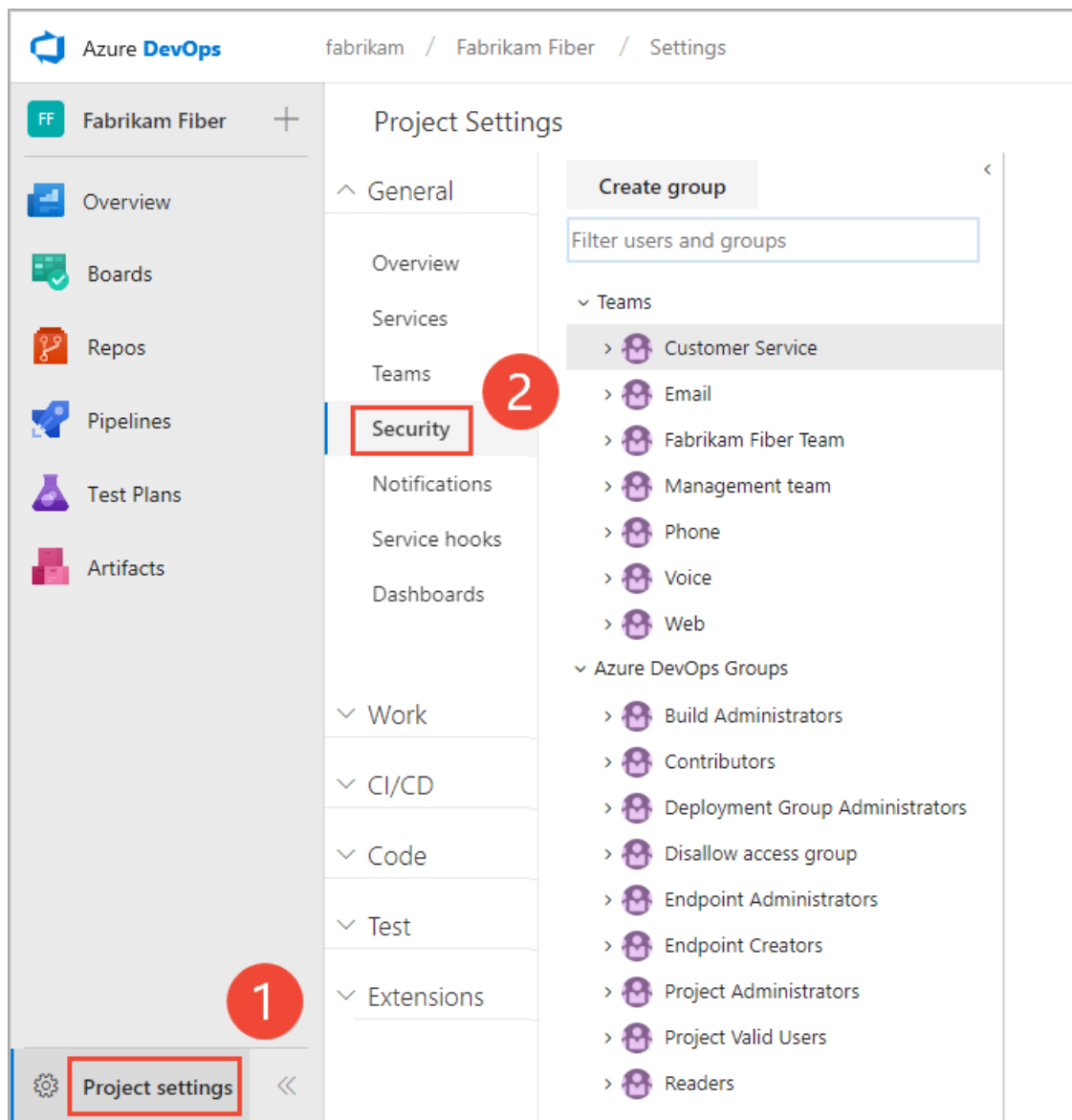
If you are adding a user to Azure DevOps for the first time, see [Add account users for Azure DevOps](#).

NOTE

Choose **Previous navigation** when you see a top-level blue bar. Choose **New navigation** if you see a vertical sidebar or if you enabled the **New Navigation** preview feature. The vertical sidebar, along with other navigational features, is enabled when the **New Navigation** preview feature has been enabled for the signed-in user or the organization. To learn how to use the web portal effectively, see [Web portal navigation](#).

For on-premises TFS, choose **Previous Navigation** for guidance.

- [New navigation](#)
 - [Previous navigation](#)
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 **Security**.



3. Under **Groups**, choose one of the following:

- To add users who will require read-only access to the project, choose **Readers**.
- To add users who will contribute fully to this project or who have been granted stakeholder access, choose **Contributors**.
- For users who will need to administrate the project, choose **Project Administrators**. To learn more, see [Set permissions at the project-level or project collection-level](#).

4. Next, choose the **Members** tab.

Here we choose the **Contributors** group.

Create group

Filter users and groups

- Teams
 - Customer Service
 - Fabrikam Fiber Team
 - Management team
 - Phone
 - Voice
 - Web
- Azure DevOps Groups
 - Build Administrators
 - Contributors**
 - Project Administrators
 - Project Valid Users
 - Readers

Fabrikam Fiber > Contributors | Edit...

Permissions **Members** Member of

+ Add... | Refresh | Search

Display Name	Username Or Scope	
Customer Service	[Fabrikam Fiber]	Remove
Fabrikam Fiber Team	[Fabrikam Fiber]	
Management team	[Fabrikam Fiber]	
Phone	[Fabrikam Fiber]	
Voice	[Fabrikam Fiber]	
Web	[Fabrikam Fiber]	
Jia-hao Tseng	fabrikamfiber9@hotmail.com	

By default, the default team group and all other teams you add to the project are included as members of the **Contributors** group. So, you can choose to add a new user as a member of a team instead, and the user would automatically inherit Contributor permissions.

TIP

Managing users is much easier [using groups](#), not individual users.

- Choose **+ Add** to add a user or a user group.
- Type the name of the user account into the text box. You can type several identities into the text box, separated by commas. The system will automatically search for matches. choose the match(es) that meet your requirements.

Add users and groups

To add users or groups to this group, just type their sign-in addresses or group aliases

User or group

Chris

Christie Church
fabrikamfiber1@hotmail.com

Showing 1 result

Save changes **Cancel**

NOTE

The first time you add a user or group to Azure DevOps or TFS, you can't browse to it or check the friendly name. After the identity has been added, you can just type the friendly name.

7. In **Identities**, specify the name of the user or group you want to add.
8. Depending on the user, you might want to customize their permissions for other functionality in the project, such as [areas and iterations](#) or [shared queries](#).

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](#).


New navigation isn't supported on TFS at this time. Choose **Previous navigation** for guidance.

Add users to a team

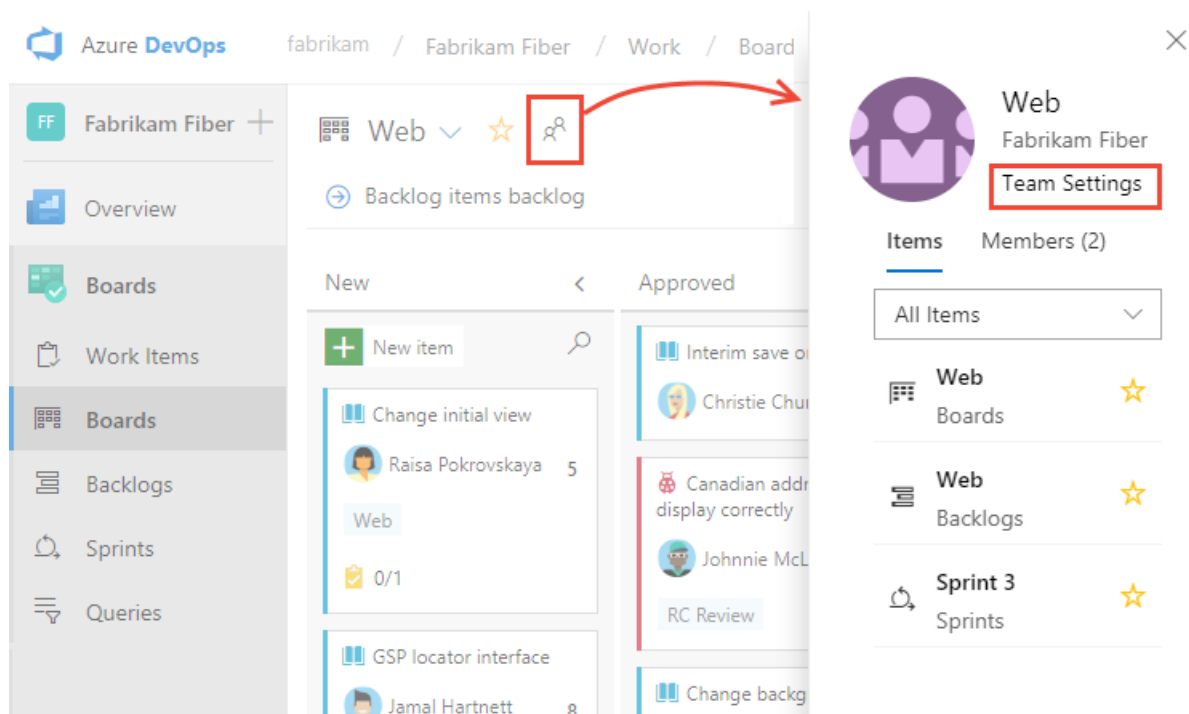
Several Agile tools, like capacity planning, team alerts, and dashboard widgets are team-scoped. That is, they automatically reference the user accounts added as members of a team to support planning activities or sending alerts. To learn more, see [About teams and Agile tools](#).

- [New navigation](#)
- [Previous navigation](#)

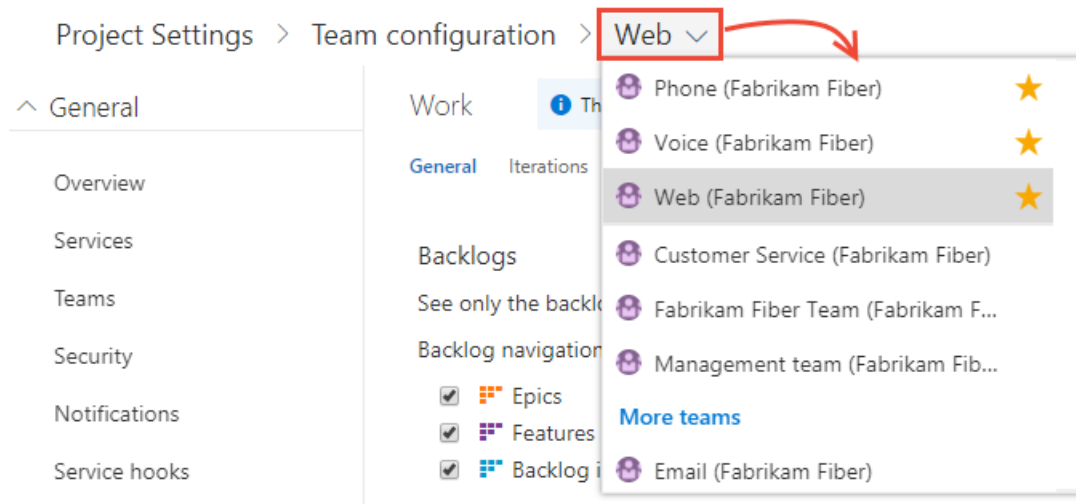
You add team members from **Project Settings>Work>Team configuration**. You can quickly navigate to it from a team work tracking backlog, board, or dashboard.

1. Open a backlog or board for a team and choose the  team profile icon. Then choose **Team Settings**.

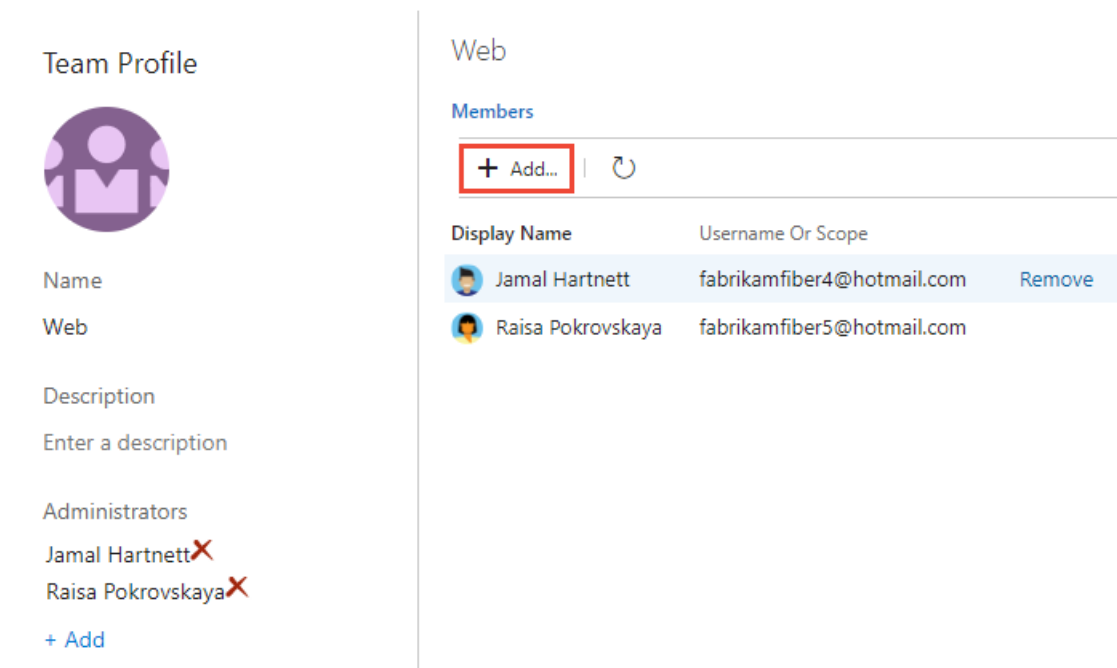
Here we open the Board for the Web team and from there the team profile.



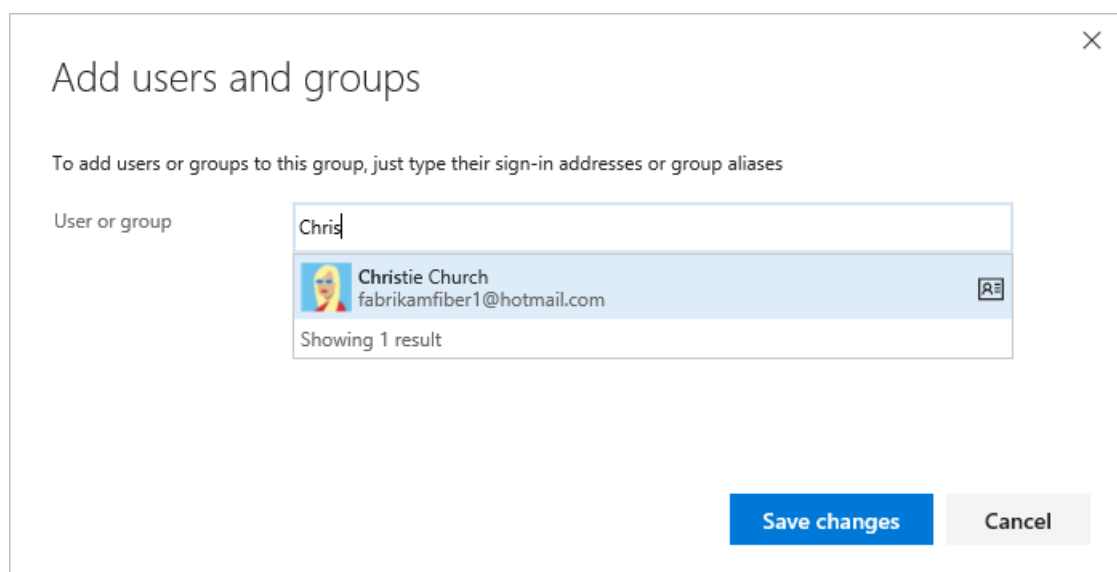
2. If you need to switch the team context, use the team selector within the breadcrumbs.



3. Choose **Add**.



4. Enter the sign-in addresses or display name for each account you want to add. Add them one at a time or all at the same time. You can type several identities into the text box, separated by commas.



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 type 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 remove members, return to this page, highlight the user name and choose **Remove**.

Team Profile



Name


Web

Description

Enter a description

Administrators

Jamal Hartnett 

Raisa Pokrovskaya 

[+ Add](#)

Web

Members



Add...



membership

Display Name

Username Or Scope



Christie Church

fabrikamfiber1@hotmail.com

[Remove](#)



Chuck Reinhart

fabrikamfiber3@hotmail.com



Helena Petersen

fabrikamfiber8@hotmail.com



Jamal Hartnett

fabrikamfiber4@hotmail.com



Raisa Pokrovskaya

fabrikamfiber5@hotmail.com

NOTE

To remove a team administrator as a team member, you must first remove them as an administrator.

6. To add an account as a team administrator, choose **Add** located in the Team Profile page. For details, see [Add a team administrator](#).

New navigation isn't supported on TFS at this time. Choose **Previous navigation** for guidance.

Add users or groups to an access level

For on-premises TFS, 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 SharePoint or SQL Server Reports (TFS only)

If your TFS deployment is integrated with a SharePoint product or SQL Server Reports, you'll need to manage membership for those products separately from their websites.

- [Set SharePoint site permissions](#)
- [Grant permissions to view or create SQL Server reports in TFS](#)

Next steps

[Add administrators or set permissions at the project or collection level](#)

To view permissions for yourself or another user, see [View permissions](#).

Related articles

You can also control access to projects, version control, build, and work items.

- [Set Git or TFVC repository permissions](#)
- [Set Git branch permissions](#)
- [Set build and release permissions](#)
- [Set permissions and access for work tracking](#)

Team administrator role and permissions

9/10/2018 • 2 minutes to read • [Edit Online](#)

Azure DevOps Services | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

While most permissions are governed by belonging to a security group or defined at the object level, team settings are managed by the team administrator role.

For each team that you add, you can assign one or more team members as administrators. The team admin role isn't a group with a set of defined permissions. Instead, the team admin role is tasked with managing and configuring [team Agile tools](#).

In addition to team administrators, all members of the Project Administrators and Project Collection Administrators groups can manage settings for all teams as well as add team administrators.

Set team defaults

Team administrators can select the area paths and iteration paths associated with their team. These settings affect a number of Agile tools available to the team.

These include the following associations made for each team:

- **Select team area paths**
Can select the default area path(s) associated with the team. These settings affect a number of 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 a number of Agile tools available to the team.

For details, see [Set team defaults](#).

Enable team backlogs and configure how bugs are managed on backlogs and boards

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, they can choose whether bugs are treated similar to user stories and requirements or as tasks.

For details, see these topics:

- [Select backlog levels for your team](#)
- [Set your team's preferences for tracking bugs](#).

Customize Kanban boards

Team administrators can fully customize the team's Kanban boards associated with the product and portfolio backlogs. This includes the following elements:

- [Cards: Fields](#)
- [Cards: Styles](#)
- [Cards: Tag colors](#)

- [Cards: Annotations](#)
- [Cards: Tests](#)
- [Board: Columns](#)
- [Board: WIP limits](#)
- [Board: Split columns](#)
- [Board: Swimlanes](#)
- [Board: Card reordering](#)
- [Board: Definition of Done](#)
- [Charts: Cumulative flow](#)

Add and manage team dashboards

Team administrators can add, configure, and manage permissions for team dashboards. For details, see [Add and manage dashboards](#).

Set working days off

Sprint planning and tracking tools automatically consider days off when calculating capacity and sprint burndown. Team admins can choose which days are non-working days through the team's Settings dialog. For details, see [Set working days](#).

Manage team alerts

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. For details, see [Manage team alerts](#).

NOTE

There is no UI associated with managing alert permissions.

NOTE

There is no UI associated with managing alert permissions. Instead, you can use **TFSecurity** to manage alerts in TFS.

Create and manage team rooms

Team administrators can add users and events to team rooms, and add team rooms. Team rooms are chat rooms limited to team members. For details, see [Collaborate in a team room](#).

NOTE

Team Rooms are deprecated for TFS 2018 and later versions as described in [Deprecation of team rooms](#) blog post. Several good solutions are available that integrate well with TFS that support notifications and chat, such as [Microsoft Teams](#) and [Slack](#).

Related articles

- [Permissions and access for work tracking](#)
- [Add teams](#)
- [Add a team administrator](#)

Quickstart: Get started as an administrator

9/10/2018 • 5 minutes to read • [Edit Online](#)

Azure DevOps Services | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

With most Azure DevOps Services or Team Foundation Server (TFS) applications, you can start using any service and configure resources as you go. No up front work is required. Most settings define defaults.

As an organization owner or a project admin, there are a few items you may want to attend to at the start, to ensure a smooth operational experience. If you own a large organization, you'll want to consider additional tasks to structure your projects to support multiple teams or software development apps.

Add users

The first task is to ensure that all members of your organization or group are added to your Azure DevOps organization and projects. For small groups, using [Microsoft Accounts](#) to add users to your organization and projects works fine.

Larger organizations will want to consider using Azure Active Directory to keep the maintenance of managing permissions and user access. To learn more, see:

- [Add organization users for Azure DevOps](#)
- [Manage user access through Azure Active Directory](#)

The first task is to ensure that all members of your organization or group are added to your organization and project.

Larger organizations will want to consider using Active Directory to keep the maintenance of managing permissions and user access. Typically, you should install Active Directory prior to installing TFS. To learn more, see:

- [Install Active Directory Domain Services \(Level 100\)](#)
- [Step-By-Step: Setting up Active Directory in Windows Server 2016](#)

You can delegate the task to add users to an organization by adding a user with Stakeholder or Basic access to the [Project Collection Administrators group](#).

Grant or restrict permissions

Access to features and functions is controlled by access level assignments and permissions. We recommend you review the following articles to understand the core concepts involved:

- [About security and identity](#)
- [About permissions and groups](#)
- [About security roles](#)
- [About access levels](#)
- [Default permissions and access](#)

If you decide you want to delegate specific tasks to others, than you'll want to add them to a built-in or custom security group, or add them to a specific role. To learn more, see:

- [Grant or restrict access to select features and functions](#)
- [Set permissions at the project-level or project collection-level](#)

Share your project vision, support collaboration

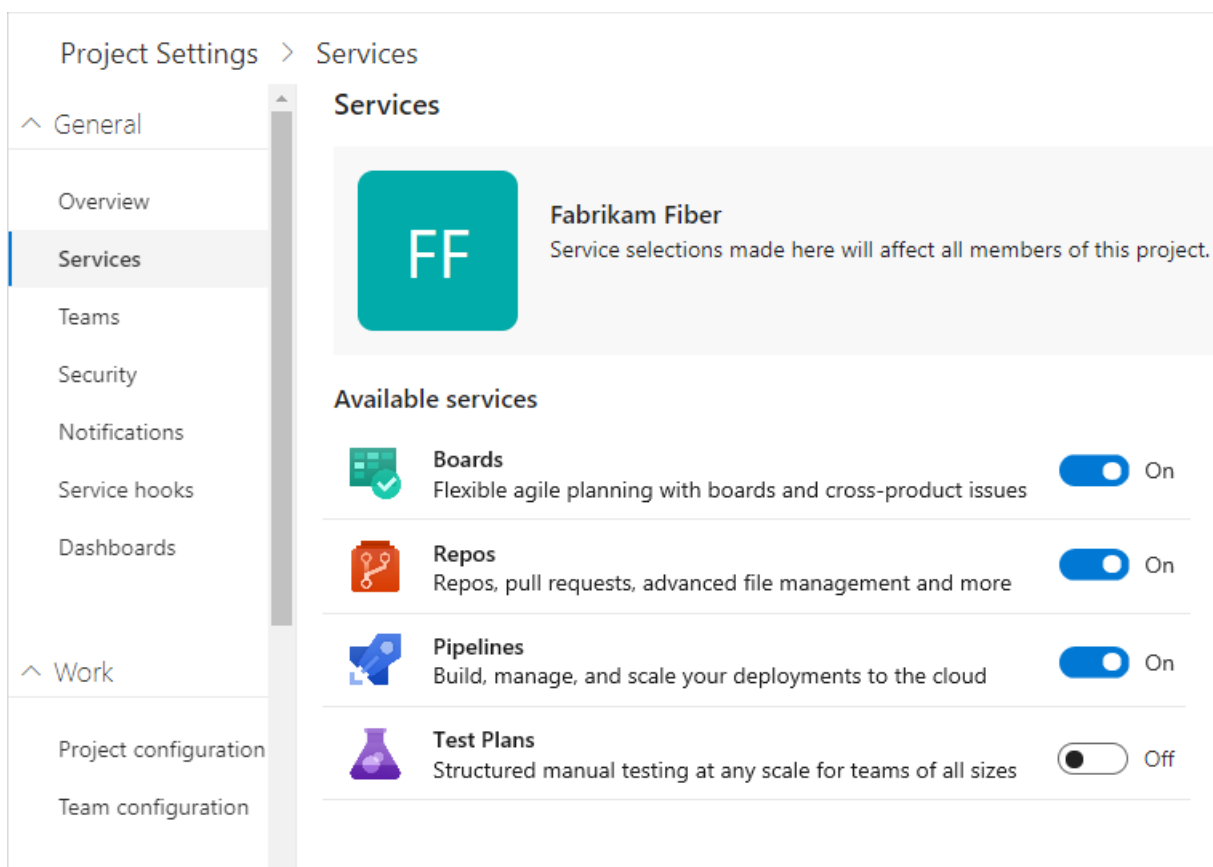
Each project has a summary page where you can share information through README files or by pointing to a project Wiki. To orient users who are new to your project and share established processes and procedures, we recommend that you [set up your project summary page](#) or [provision a Wiki](#).

Each project has a summary page where you can share information through README files. To orient users who are new to your project and share established processes and procedures, we recommend that you [set up your project summary page](#).

Remove unused services from the user interface

To simplify the web portal user interface, you can disable select services. If you use a project to only log bugs, then you can remove all services except for **Boards**.

For example, here we show that **Test Plans** has been disabled.



Set code, test, and other policies

There are several policies you can set to support collaboration across your teams, secure your projects, and to automatically remove files no longer needed. You'll want to review the following articles to set policies.

- [Change application access policies for your organization](#)
- [Manage branch policies](#)
- [Add TFVC check-in policies](#)
- [Set build and release pipeline retention policies](#)
- [Set test retention policies](#)
- [Manage branch policies](#)
- [Add TFVC check-in policies](#)

- [Set build and release pipeline retention policies](#)
- [Set test retention policies](#)

Define area and iteration paths for work tracking

If you support several products or feature areas, you can assign work items by feature area by setting up [area paths](#). To assign work items to specific time intervals, also known as sprints, you'll want to configure [iteration paths](#). To use the Scrum tools—sprint backlogs, taskboards, team capacity—you need to configure several sprints. For an overview, see [About areas and iteration paths](#).

Iterations

Areas

Create and manage the iterations for this project. These it for iteration planning (sprint planning). [Click Here](#)

To access the default team's iteration settings, [Click Here](#).

NewNew child

Iterations	Start Date	End Date
<div> <div>Fabrikam Fiber</div> <div> <div>Release 1</div> <div> <div>Sprint 1</div> <div>6/11/2018</div> <div>6/29/2018</div> </div> <div> <div>Sprint 2</div> <div>7/2/2018</div> <div>7/20/2018</div> </div> <div> <div>Sprint 3</div> <div>7/16/2018</div> <div>8/3/2018</div> </div> <div> <div>Sprint 4</div> <div>7/23/2018</div> <div>8/10/2018</div> </div> <div> <div>Sprint 5</div> <div>9/17/2018</div> <div>10/5/2018</div> </div> <div> <div>Sprint 6</div> <div>10/29/2018</div> <div>11/16/2018</div> </div> </div> <div>Release 2</div> <div>Release 3</div> </div>		

Iterations

Areas

Create and manage the areas for this project. These a the team's backlog and what work items the team is r

To access the default team's area settings, [Click Here](#).

NewNew child

Areas	Teams
<div> <div>Fabrikam Fiber</div> <div> <div>Customer Service</div> <div>Phone</div> <div>Voice</div> <div>Web</div> </div> </div>	<div> <div>Fabrikam Fiber Team</div> <div>Customer Service Team</div> <div>Fabrikam Fiber Team, Phone</div> <div>Voice</div> <div>Fabrikam Fiber Team, Web</div> </div>

Customize work tracking processes

While you and your teams can start using all work tracking tools immediately after you create a project, oftentimes one or more users will want to customize the experience to meet one or more business needs. While you can customize the process easily through the user interface, you'll want to establish a methodology for who will manage the updates and evaluate requests.

NOTE

By default, users granted Basic and Stakeholder access 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, then you can set their permissions at the organization level to **Deny**.

To learn more, see:

- [About process customization and inherited processes](#)
- [Customize a project](#)
- [Add and manage processes](#)

Customize work tracking processes

While you and your teams can start using all work tracking tools immediately after you create a project, oftentimes one or more users will want to customize the experience to meet one or more business needs. You'll want to establish a methodology for who will manage the updates and evaluate requests.

To learn more, see [On-premises XML process model](#).
















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 notifications](#), managed by a team administrator
- Project notifications, managed by a member of the Project Administrators group
- [Organization/collection-level notifications](#), managed by a member of the Project Collection Administrators group

If users believe they are getting too many notifications, they can [opt out of a subscription](#).

🔔 Notifications > Mine | [+ New subscription](#) [? Help](#)

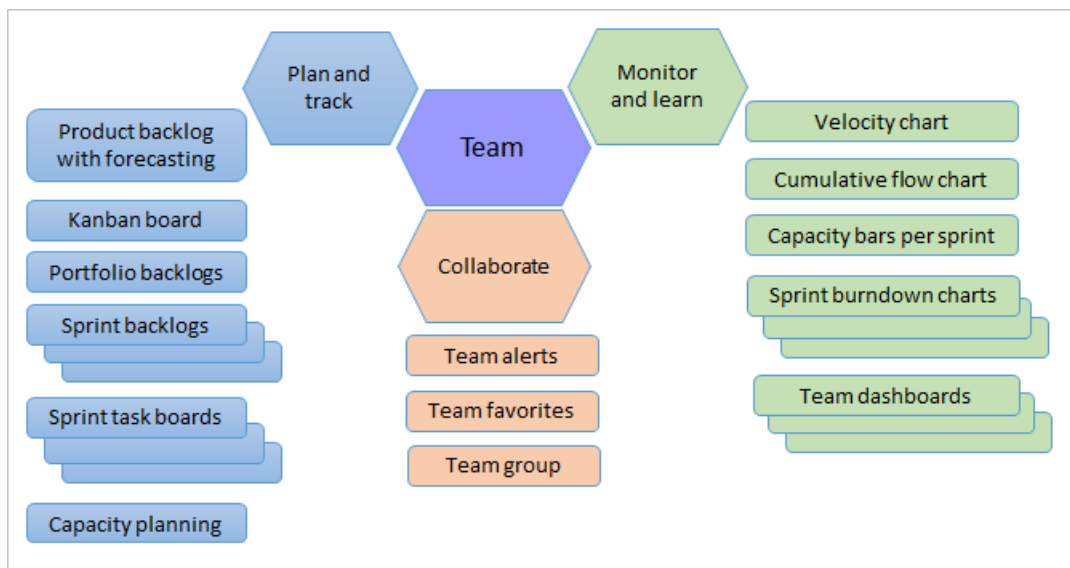
Description	Type	Notifies	State
Build			
 Build completes Notifies you when a build you queued or that was queued for you compl...	 Build completed (any project)	 You	<input checked="" type="checkbox"/> On
Code (Git)			
 Pull request reviewers added or removed Notifies you when you are added to a pull 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

Configure an SMTP server

In order for team members to receive notifications, [you must configure an SMTP sever](#).

Add teams to scale your organization

We recommend you add teams as your organization grows. Each team gets [access to their own set of Agile tools](#) which they can customize.



To learn more, see the following:

- [About projects and scaling your organization](#)
- [Add a team, move from one default team to several teams](#)
- [Add team administrator](#)

Install and manage extensions

An extension is an installable unit that contributes new capabilities to your projects. You can find extensions from within the [Visual Studio Marketplace](#) to support planning and tracking of work items, sprints, scrums, etc.; build and release flows; code testing and tracking; and collaboration among team members.

For example, to support [code search](#), install the [Code Search extension](#).

You'll want to let your users know about extensions and that they can [request an extension](#). To install and manage extensions, you must be an organization owner, be a member of the Project Collection Administrators group, or be added to the [Manager role for extensions](#).

Set up billing

All organizations can add up to five users with Basic access and unlimited users with Stakeholder access. If you need to add more users or pay for additional services or extensions, then you'll want to [set up billing](#).

Next steps

[Manage projects](#)

Related articles

- [Security & identity](#)
- [Manage organizations](#)
- [All settings](#)
- [Manage projects](#)
- [Security & identity](#)
- [Manage organizations](#)
- [All settings](#)
- [TFS administration](#)

Install free extensions for Azure DevOps

9/11/2018 • 2 minutes to read • [Edit Online](#)

Azure DevOps Services

To add new features and capabilities to your Azure DevOps organization, install extensions from the [Visual Studio Marketplace](#). You can install [free](#), [preview](#), or [paid](#). In this quickstart, you learn how to install a free extension.

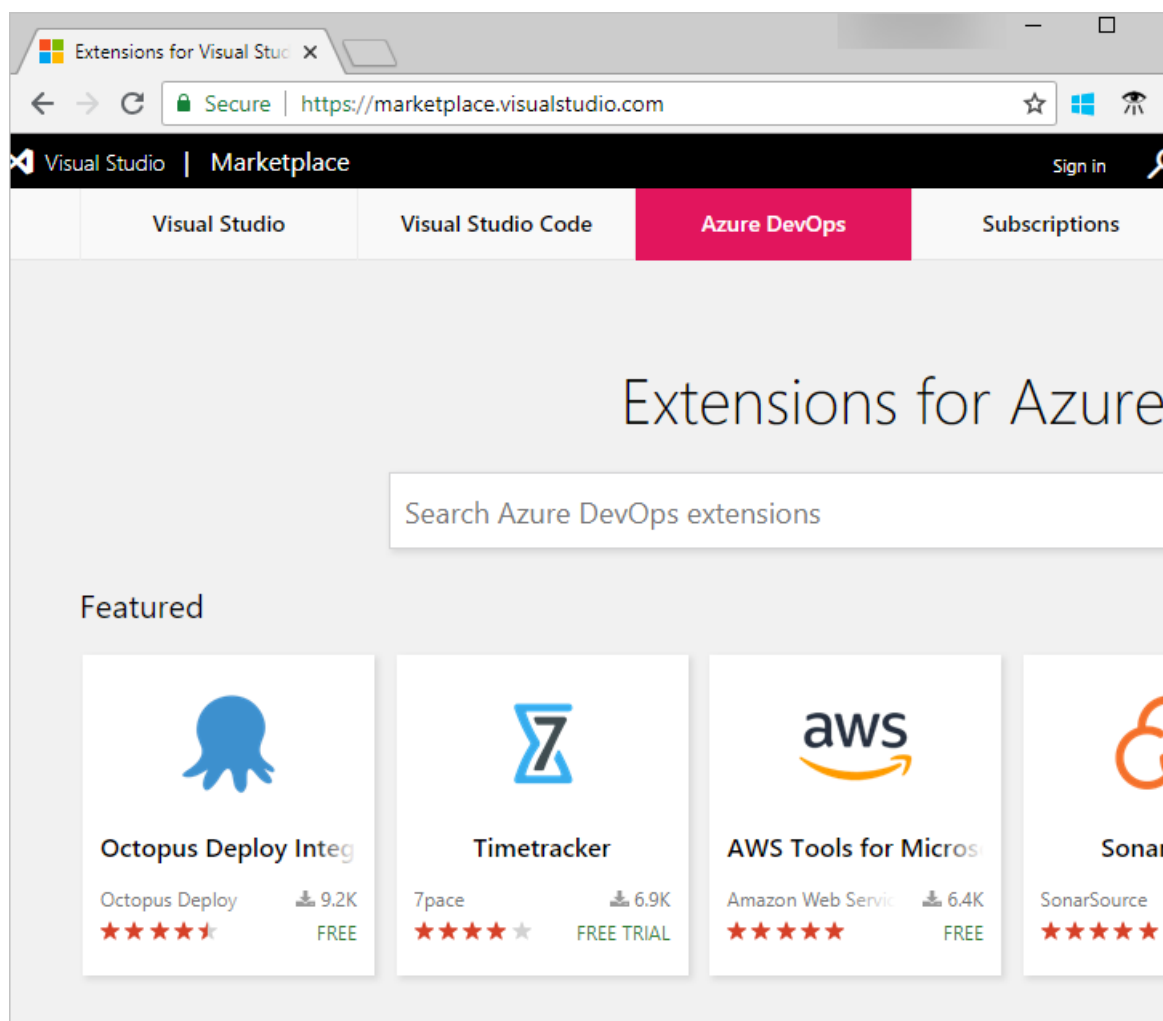
To learn about building your own Azure DevOps extensions, see [developing](#) and [publishing](#) extensions.

Prerequisites

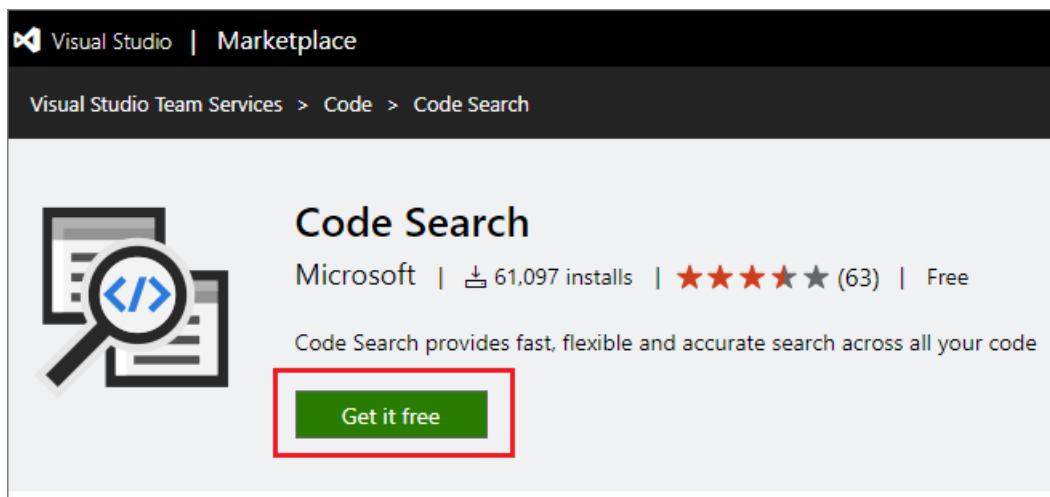
- Only Azure DevOps [project collection administrators](#) or [organization owners](#) can install extensions. If you don't have permissions, you can [request extensions](#) instead.
- Private extensions must be shared with your Azure DevOps organization to be installed. Check out the [publishing documentation](#) for information on how to share private extensions.

Install the extension

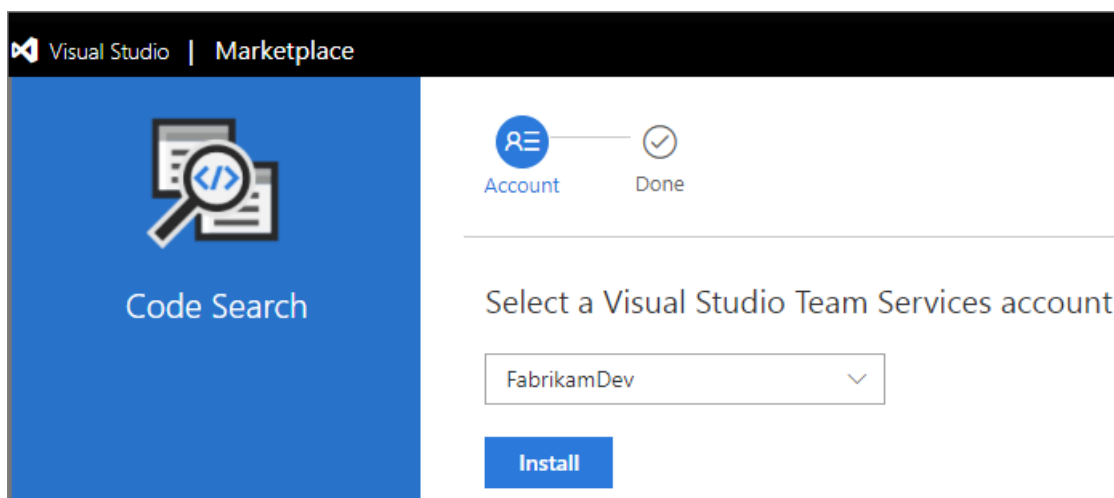
1. Sign in to the [Visual Studio Marketplace > Azure DevOps](#).



2. Find and select the extension that you want to install. For this quickstart, we select [Code Search](#).
3. Choose **Get it free**.

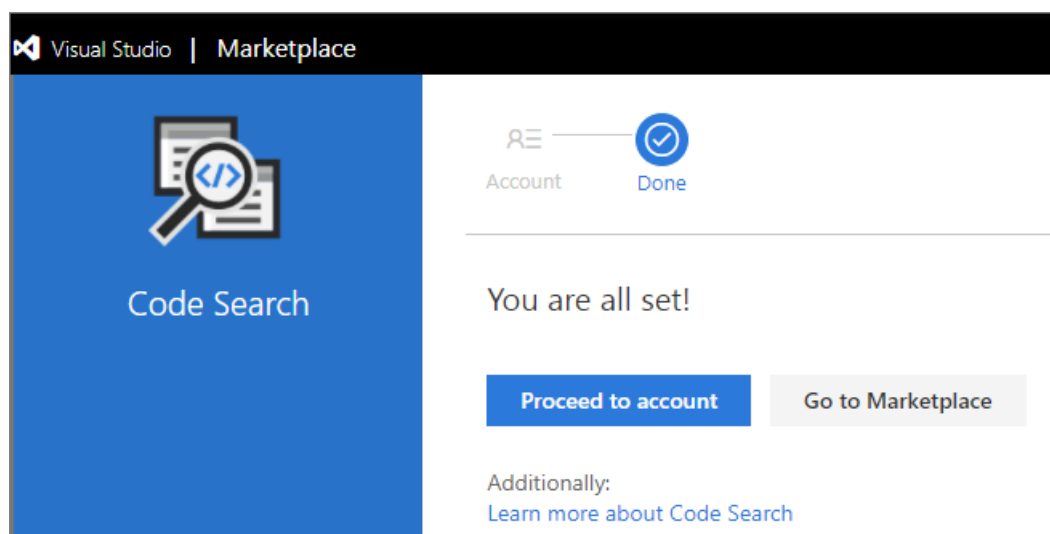


4. Select your Azure DevOps organization and select **Install** to install the extension.



- [Why don't I see any Azure DevOps organizations?](#)
- [Why can't I install this extension?](#)

Your extension is now installed! You can now go to your Azure DevOps organization to use your extension. Also, tell your team about this extension, so they can start using its capabilities too.



Next steps

[Assign paid extensions](#)

Related articles


Find answers to common problems on the [troubleshooting](#) page.

Set personal or team favorites

9/10/2018 • 8 minutes to read • [Edit Online](#)

Azure DevOps Services | TFS 2018 | TFS 2017.1

By favoriting an artifact or view, you can quickly access it. An artifact is a specific container, definition, or configuration of a feature or tool—such as a project, repository, build pipeline, or query. As your code base, work tracking efforts, developer operations, and organization grows, you'll want to be able to quickly navigate to those artifacts of interest to you and your team. Setting favorites allows you to do just that.

You can set favorites for yourself or your team. 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 clicking 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 artifact.

You can set favorites for the following artifacts from the indicated page.

NOTE

Choose **Previous navigation** when you see a top-level blue bar. Choose **New navigation** if you see a vertical sidebar or if you enabled the **New Navigation** preview feature. The vertical sidebar, along with other navigational features, is enabled when the **New Navigation** preview feature has been enabled for the signed-in user or the organization. To learn how to use the web portal effectively, see [Web portal navigation](#).

For on-premises TFS, choose **Previous Navigation** for guidance.

- [New navigation](#)
- [Previous navigation](#)

SERVICE/AREA	ARTIFACTS	PAGE
Organization & projects	Projects	Projects (personal favorites only)
Dashboard	Team dashboards	Overview>Dashboards
Code	Repositories	Repos>Branches (personal favorites only)
Work, Agile tools for a team	Backlogs, boards, sprint backlogs, sprint taskboards	Boards>Boards or Backlogs , or Sprints
Work	Queries	Boards>Queries
Work	Delivery plans	Boards>Plans (personal favorites only, requires installation of the Delivery Plans extension)
Build and Release	Pipeline definitions	Pipelines>Builds or Releases
Test	Test plans	Test Plans>Test Plans (personal favorites only)

New navigation isn't supported on TFS at this time. Choose **Previous navigation** for guidance.

Favorite a project or team

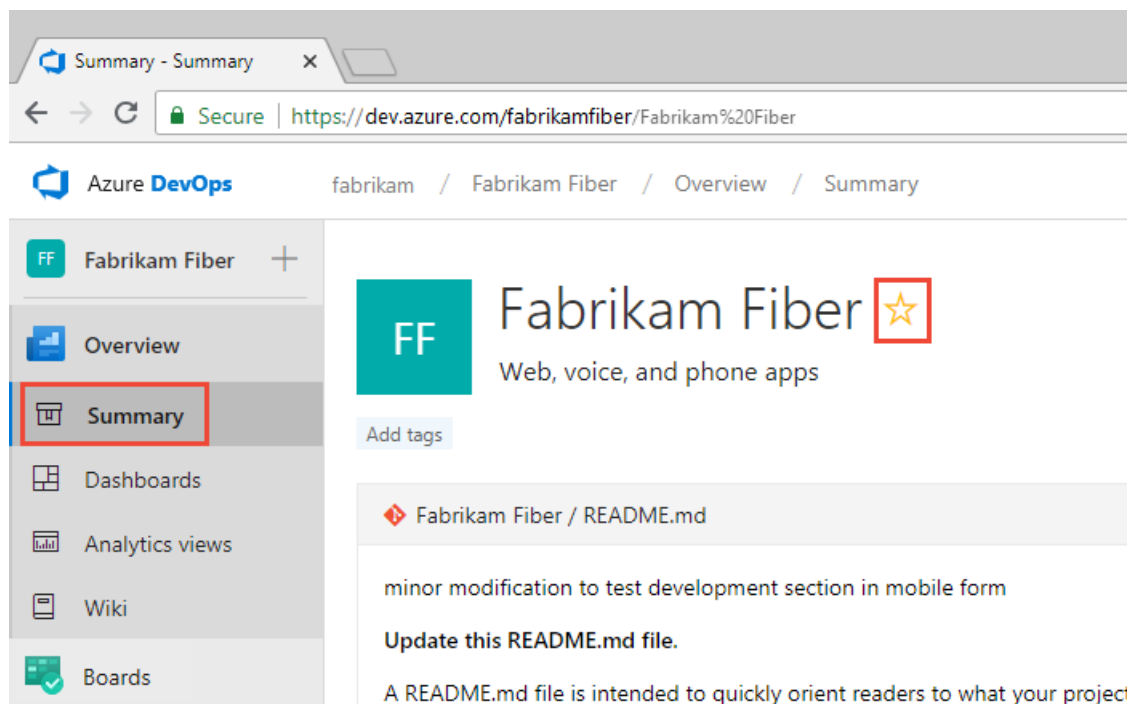
NOTE

Choose **Previous navigation** when you see a top-level blue bar. Choose **New navigation** if you see a vertical sidebar or if you enabled the **New Navigation** preview feature. The vertical sidebar, along with other navigational features, is enabled when the **New Navigation** preview feature has been enabled for the signed-in user or the organization. To learn how to use the web portal effectively, see [Web portal navigation](#).

For on-premises TFS, choose **Previous Navigation** for guidance.

- [New navigation](#)
- [Previous navigation](#)

1. To favorite a project, open the project **Summary** page and choose the ★ star icon.



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.



3. To favorite other team artifacts, choose the 🧑 team icon, and then choose the ★ star icon next to one of the listed artifacts.



Phone
Fabrikam Fiber
Team Settings

Items Members (1)

All Items



Phone
Boards



Phone
Backlogs



Phone
Sprints



New navigation isn't supported on TFS at this time. Choose **Previous navigation** for guidance.

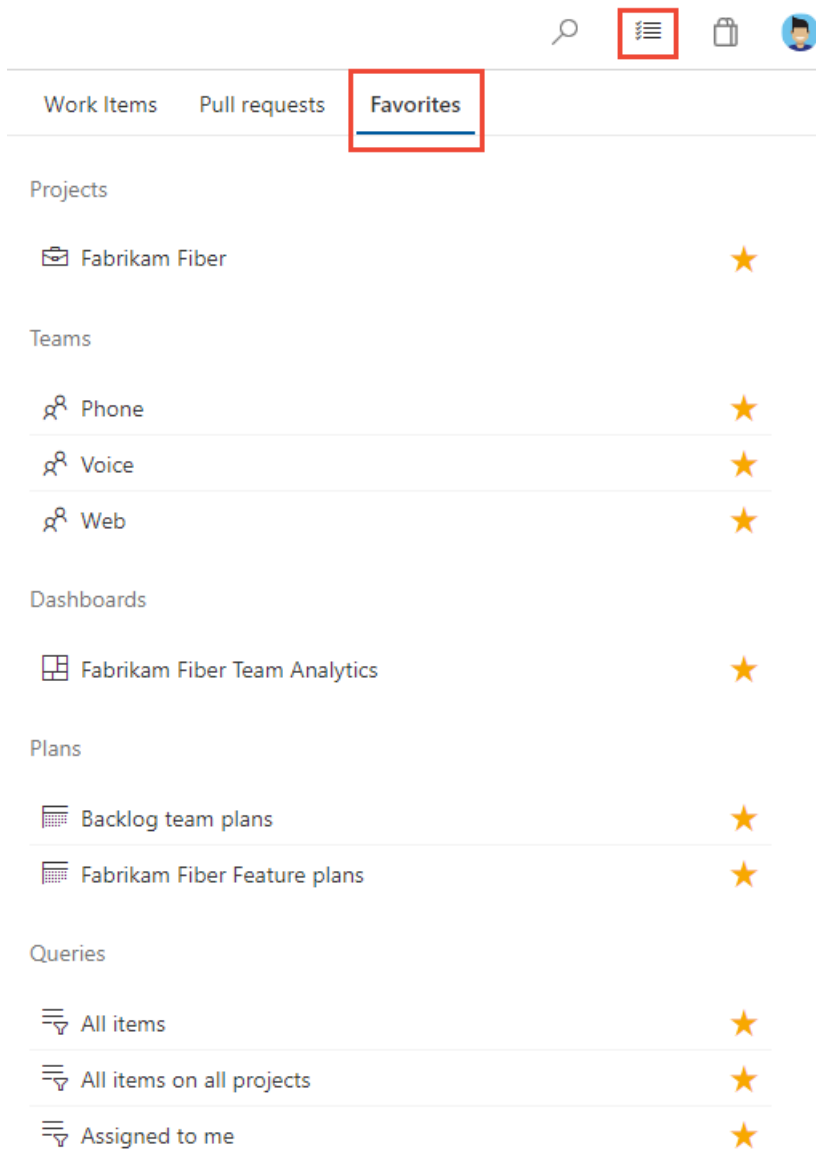
View personal favorites

NOTE

Favorite groups specific to a service are disabled if the service they depend on has been disabled. For example, if **Boards** is disabled, then the favorite groups—Plans, Boards, Backlogs, Analytics views, Sprints, and Queries item and all Analytics widgets—are disabled. To re-enable a service, see [Turn an Azure DevOps service on or off](#).

- [New navigation](#)
- [Previous navigation](#)

Access your favorites by choosing the  inbox icon, and then choosing **Favorites**.



New navigation isn't supported on TFS at this time. Choose **Previous navigation** for guidance.

Favorite a team's backlog, Kanban board, or other artifact

- [New navigation](#)
- [Previous navigation](#)

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**.

Order	Assigned To	State	Title
1	Jamal Hartnett	Committed	Hello World Web Site
2	Jamal Hartnett	Committed	Slow response on inform:
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.

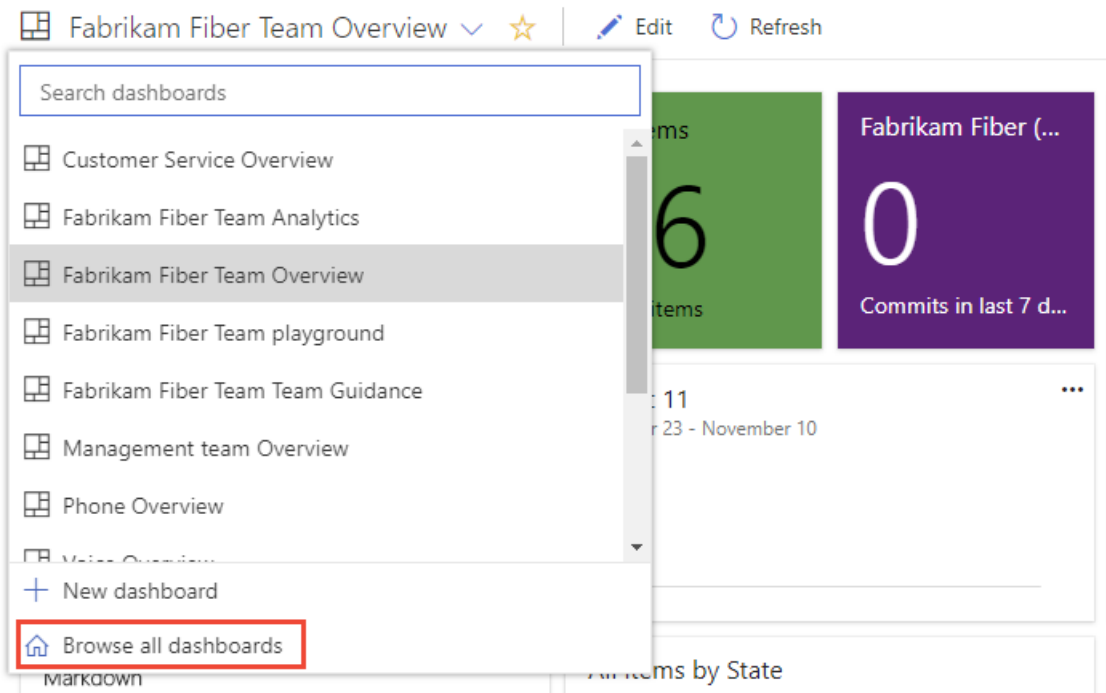
- Choose the star icon to favorite a team backlog. Favorited artifacts (favorited icon) appear appear on your **Favorites** page and towards the top of the team backlog selector menu.

New navigation isn't supported on TFS at this time. Choose **Previous navigation** for guidance.

Favorite a dashboard

- [New navigation](#)
- [Previous navigation](#)

- From **Overview>Dashboards**, open the selector and choose the **Browse all dashboards** option.



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.

Dashboards

Mine All | + New dashboard

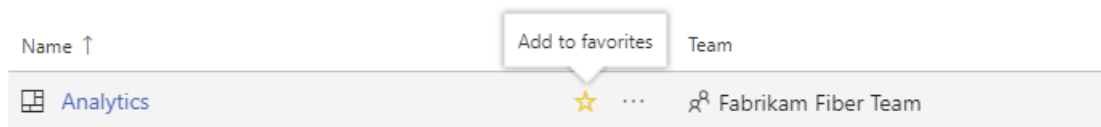
Filter dashboards Filter by team Clear

Name ↑	Team	Description
Analytics	Fabrikam Fiber Team	
Bug status	Fabrikam Fiber Team	Active bugs and bug trends
Bugs	Web	
Overview	Customer Service	
Overview	Fabrikam Fiber Team	
Overview	Management team	
Overview	Phone	
Team Guidance	Fabrikam Fiber Team	
Test	Web	
Work in Progress	Web	

TIP

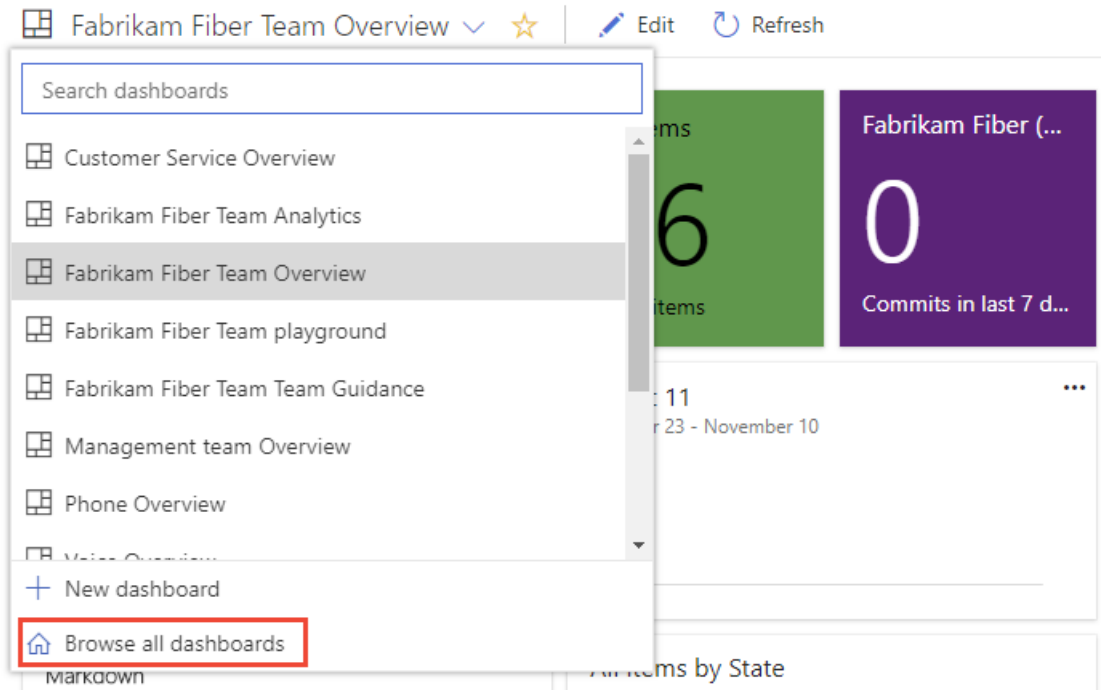
You can change the sort order of the list by choosing the column label.

3. To favorite a dashboard, hover over the dashboard and choose the .



Favoriting a dashboard will cause it to appear on your **Favorites** page and towards the top in the **Dashboards** selection menu.





















1. From **Dashboards**, open the selector and choose the **Browse all dashboards** option.



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.

Dashboards





Mine **All** | [+ New dashboard](#)

Filter dashboards			Filter by team	Clear
Name ↑	Team	Description		
 Analytics	 Fabrikam Fiber Team			
 Bug status	 Fabrikam Fiber Team	Active bugs and bug trends		
 Bugs	 Web			
 Overview	 Customer Service			
 Overview	 Fabrikam Fiber Team			
 Overview	 Management team			
 Overview	 Phone			
 Team Guidance	 Fabrikam Fiber Team			
 Test	 Web			
 Work in Progress	 Web			

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 .


Name ↑	Add to favorites		Team
 Analytics			 Fabrikam Fiber Team

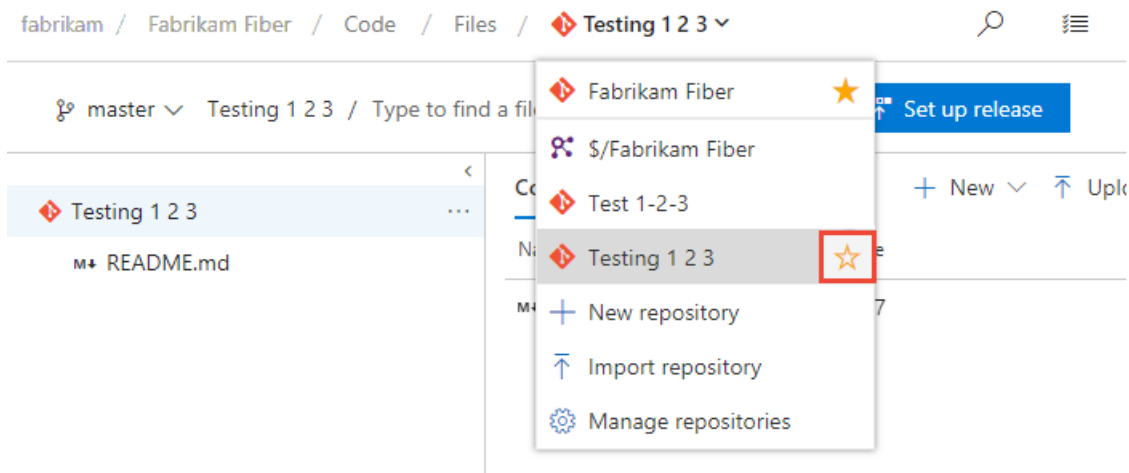
Favoriting a dashboard will cause it to appear on your **Favorites** page and towards the top in the **Dashboards** selection menu.

New navigation isn't supported on TFS at this time. Choose **Previous navigation** for guidance.

Favorite a repository

- [New navigation](#)
- [Previous navigation](#)

From any **Repos** page, open the repository selector and choose the  star icon for the repository you want to favorite.

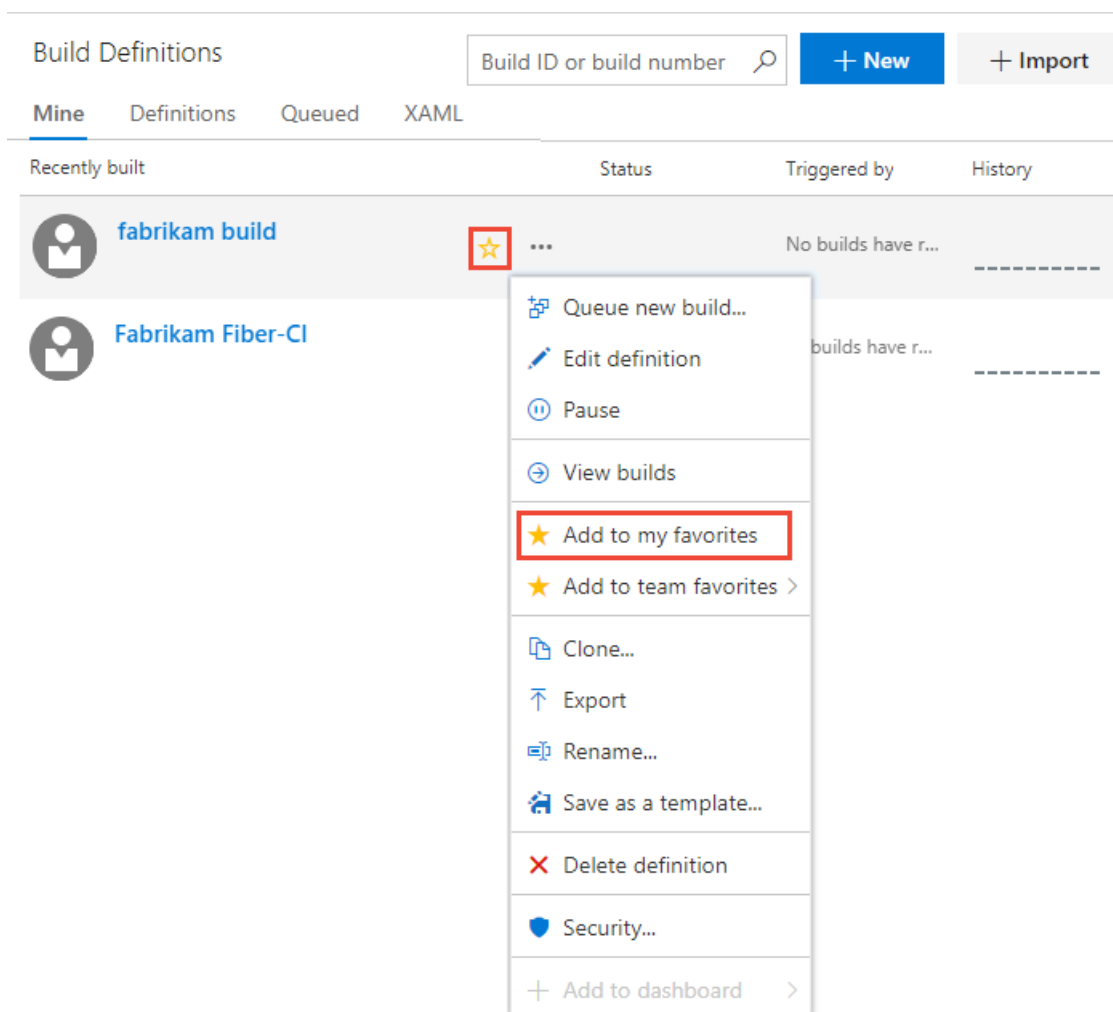


New navigation isn't supported on TFS at this time. Choose **Previous navigation** for guidance.

Favorite a build pipeline

- [New navigation](#)
- [Previous navigation](#)

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**.



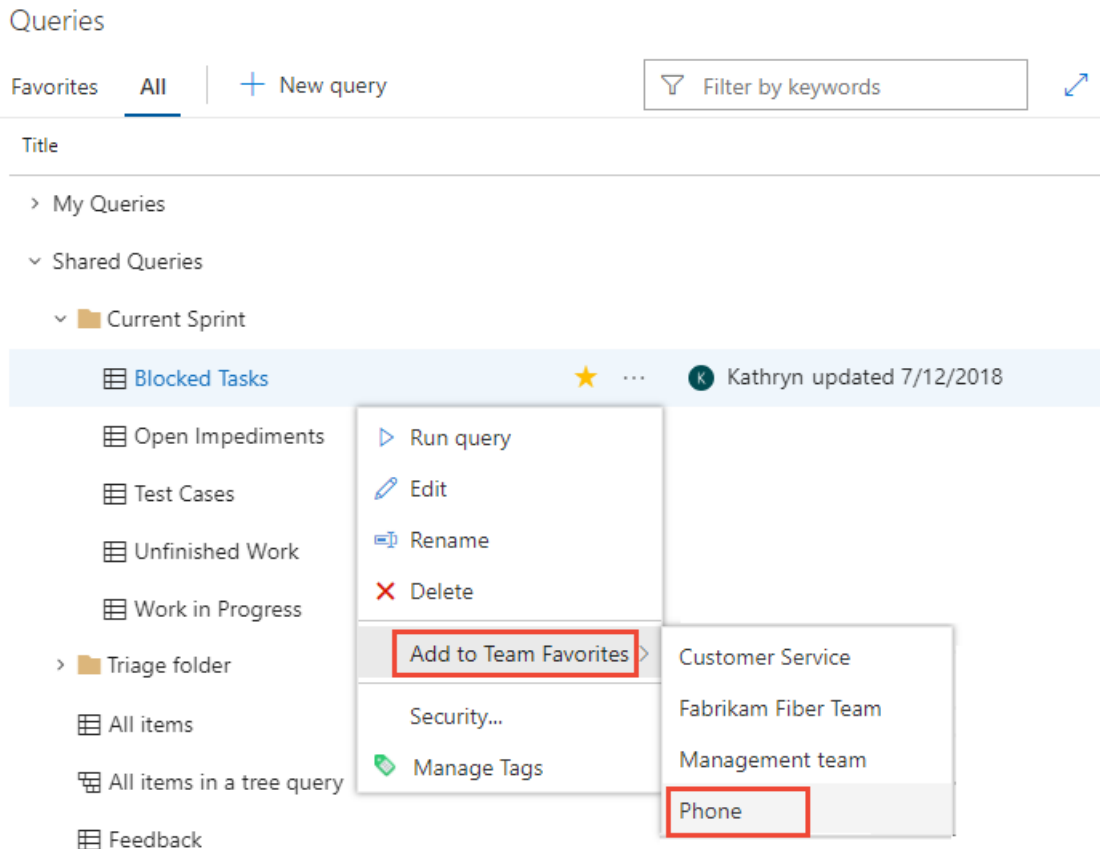
New navigation isn't supported on TFS at this time. Choose **Previous navigation** for guidance.

Favorite a shared query

- [New navigation](#)
- [Previous navigation](#)

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.



You can also set a query as a personal favorite by opening the query and choosing the ☆ star icon.

Queries > Shared Queries > All items > ☆

New navigation isn't supported on TFS at this time. Choose **Previous navigation** for guidance.

Favorite a delivery plan

To learn more about delivery plans, see [Review team Delivery Plans](#).

- [New navigation](#)
- [Previous navigation](#)

To mark a delivery plan as a favorite, open the **Boards>Plans** page and choose the ☆ star icon next to the Delivery Plan.

New navigation isn't supported on TFS at this time. Choose **Previous navigation** for guidance.

Favorite a test plan

To learn more about test plans, see [Create a test plan and test suite](#).

- [New navigation](#)
- [Previous navigation](#)

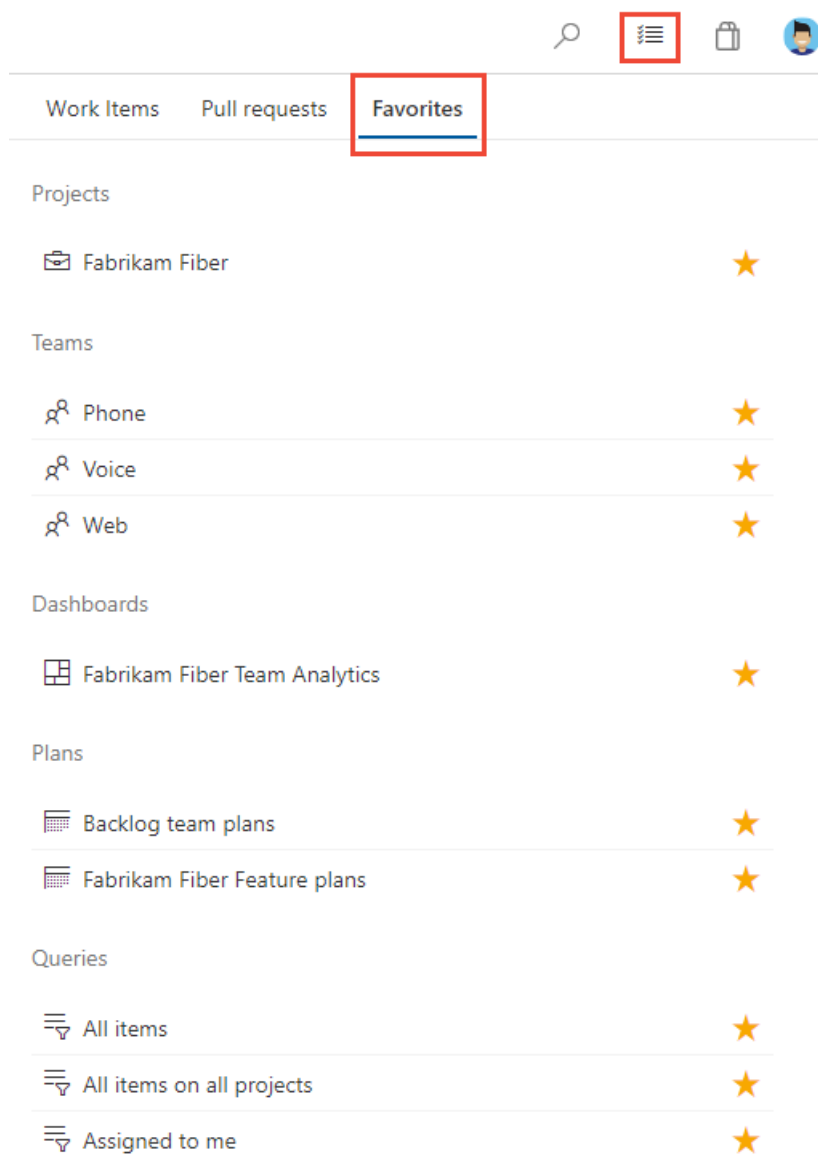
To mark a test plan as a favorite, open **Test Plans>Test Plans** and choose the ★ star icon next to a test plan from the menu that shows All test plans.

New navigation isn't supported on TFS at this time. Choose **Previous navigation** for guidance.

Unfavorite an artifact

- [New navigation](#)
- [Previous navigation](#)

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.



Similarly, you can unfavorite an artifact from the same page where you favorited it.

New navigation isn't supported on TFS at this time. Choose **Previous navigation** for guidance.

Related articles

- [Manage personal notifications](#)

- [Set your preferences](#)

Follow a work item or pull request

9/10/2018 • 3 minutes to read • [Edit Online](#)

Azure Boards | TFS 2018 | TFS 2017

NOTE

The **Follow a work item** feature is available from TFS 2017 and later versions. The **Follow a pull request** feature is available from TFS 2017.1 and later versions. To update your on-premises TFS, visit the [Visual Studio downloads page for Team Foundation Server](#).


To get notified of changes made to a work item or a pull request, you can elect to follow them.

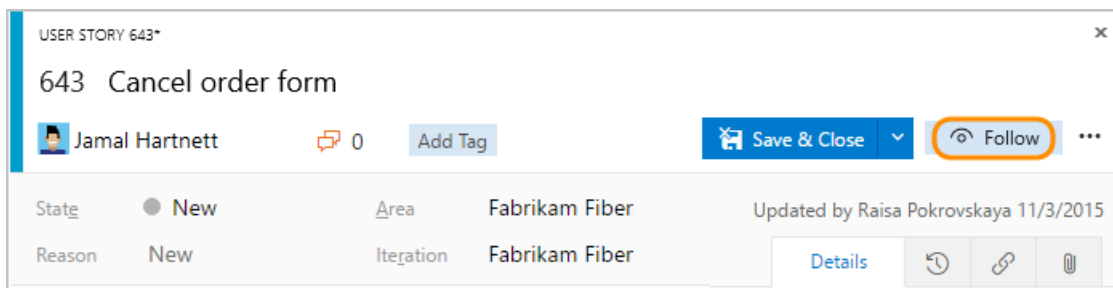
This topic shows you how to:

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

You must configure an [SMTP sever](#) in order for team members to receive notifications.

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.





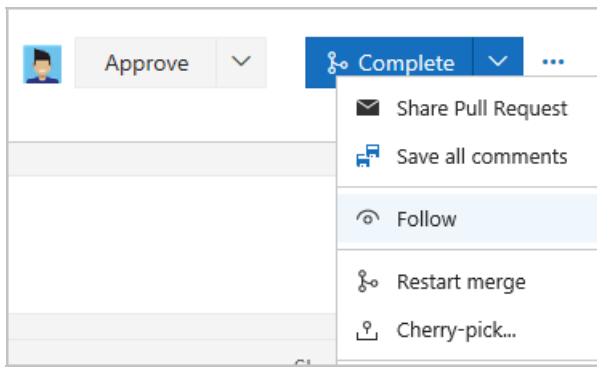
You'll only receive notifications when other members of your team modifies the work item, such as adding to the discussion, changing a field value, or adding an attachment.

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

To stop following changes, choose the  **Following** 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 modifies 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** Following icon.

Manage work items that you're following

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

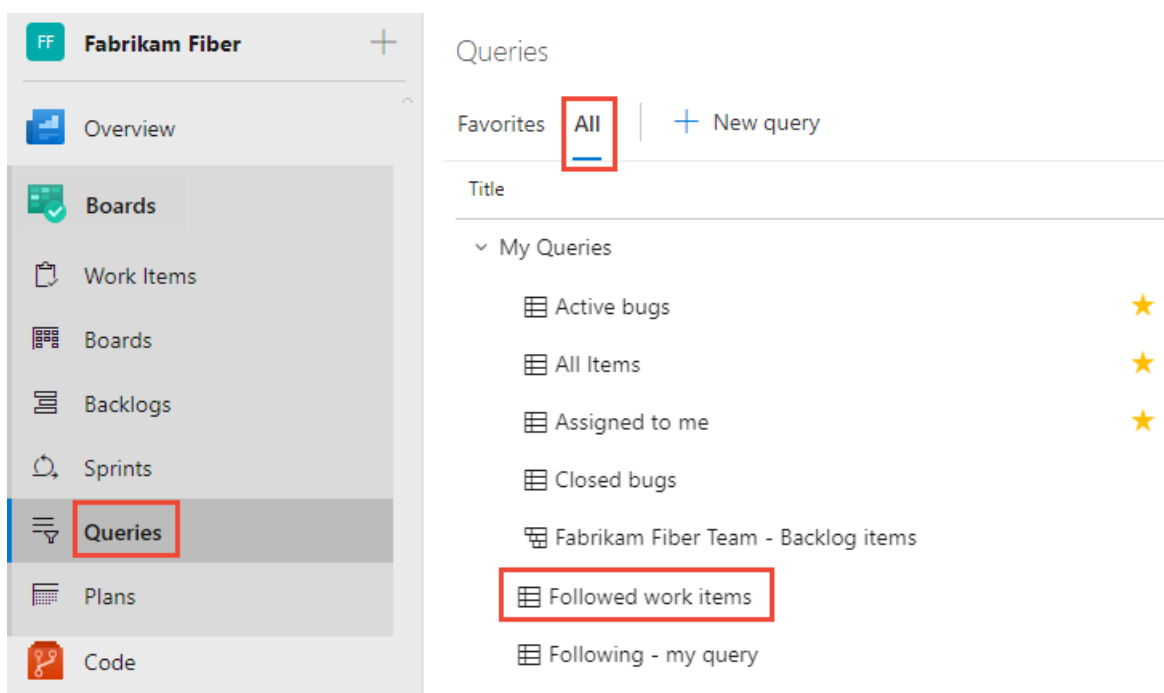
NOTE

Choose **Previous navigation** when you see a top-level blue bar. Choose **New navigation** if you see a vertical sidebar or if you enabled the **New Navigation** preview feature. The vertical sidebar, along with other navigational features, is enabled when the **New Navigation** preview feature has been enabled for the signed-in user or the organization. To learn how to use the web portal effectively, see [Web portal navigation](#).

For on-premises TFS, choose **Previous Navigation** for guidance.

- [New navigation](#)
- [Previous navigation](#)

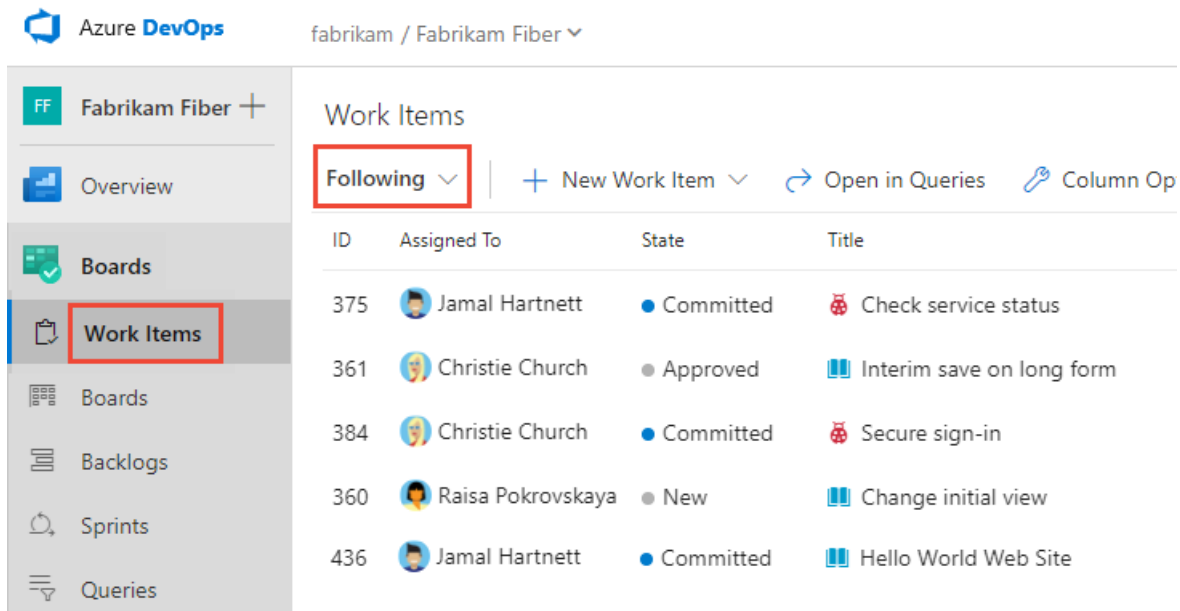
Open **Boards>Queries**, choose **All**, and under **My Queries**, choose **Followed work items**.



From this view, you can view all items you're following across all projects. Also, you can perform 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**.



Azure DevOps fabrikam / Fabrikam Fiber

FF Fabrikam Fiber +

Overview

Boards

Work Items

Boards

Backlogs

Sprints

Queries

Work Items

Following ▾ | + New Work Item ▾ ↗ Open in Queries 🔗 Column Op

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

New navigation isn't supported on TFS at this time. Choose **Previous navigation** for guidance.

Related articles

- [Manage personal notifications](#)
- [Set team 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: 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 create or modified, or a pull request is created. For details, see [Manage team notifications](#).

Get started as a Stakeholder

9/10/2018 • 10 minutes to read • [Edit Online](#)

Azure DevOps Services | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

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. Stakeholder access is one of several supported access levels. To understand the full set of features Stakeholders have access to, see [About access levels](#).

NOTE

For public projects, Stakeholder access gives users greater access to work tracking features. To learn more, see [Default roles and access for public projects](#).

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. To understand the full set of features Stakeholders have access to, see [About access levels](#).

With Stakeholder access, you can add and modify work items. 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. To understand the full set of features Stakeholders have access to, see [About access levels](#).

Use this topic to learn:

- How to sign-in to a project
- How to add a work item
- How to view the product backlog and add new work to it
- How to view work in progress on the Kanban board
- Find work assigned to you, or query for other work items

For information about working with pipelines, see these articles:

- [Build your GitHub repository](#)
- [Build OSS repositories](#)

First time signing in

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.

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

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

For example, to connect to the server named *FabrikamPrime* and project named *Contoso*, type

```
http://FabrikamPrime:8080/tfs/DefaultCollection/Contoso .
```

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

View and add work items

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

NOTE

Choose **Previous navigation** when you see a top-level blue bar. Choose **New navigation** if you see a vertical sidebar or if you enabled the **New Navigation** preview feature. The vertical sidebar, along with other navigational features, is enabled when the **New Navigation** preview feature has been enabled for the signed-in user or the organization. To learn how to use the web portal effectively, see [Web portal navigation](#).

For on-premises TFS, choose **Previous Navigation** for guidance.

- [New navigation](#)
- [Previous navigation](#)

1. (1) Check that you have selected the right project, then (2) choose **Boards>Work Items**.

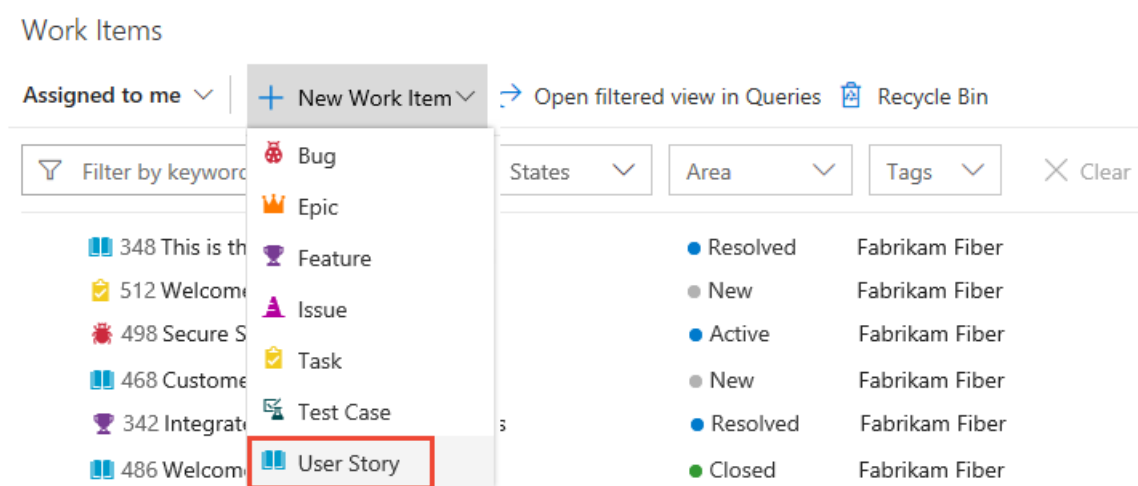
ID	Assigned To	State	Title
375	Jamal Hartnett	Committed	Check service status
361	Christie Church	Approved	Interim save on long form
384	Christie Church	Committed	Secure sign-in
360	Raisa Pokrovskaya	New	Change initial view
436	Jamal Hartnett	Committed	Hello World Web Site

2. Using the drop-down menu, you can focus on relevant items inside a project using one of the seven pivots as described next.

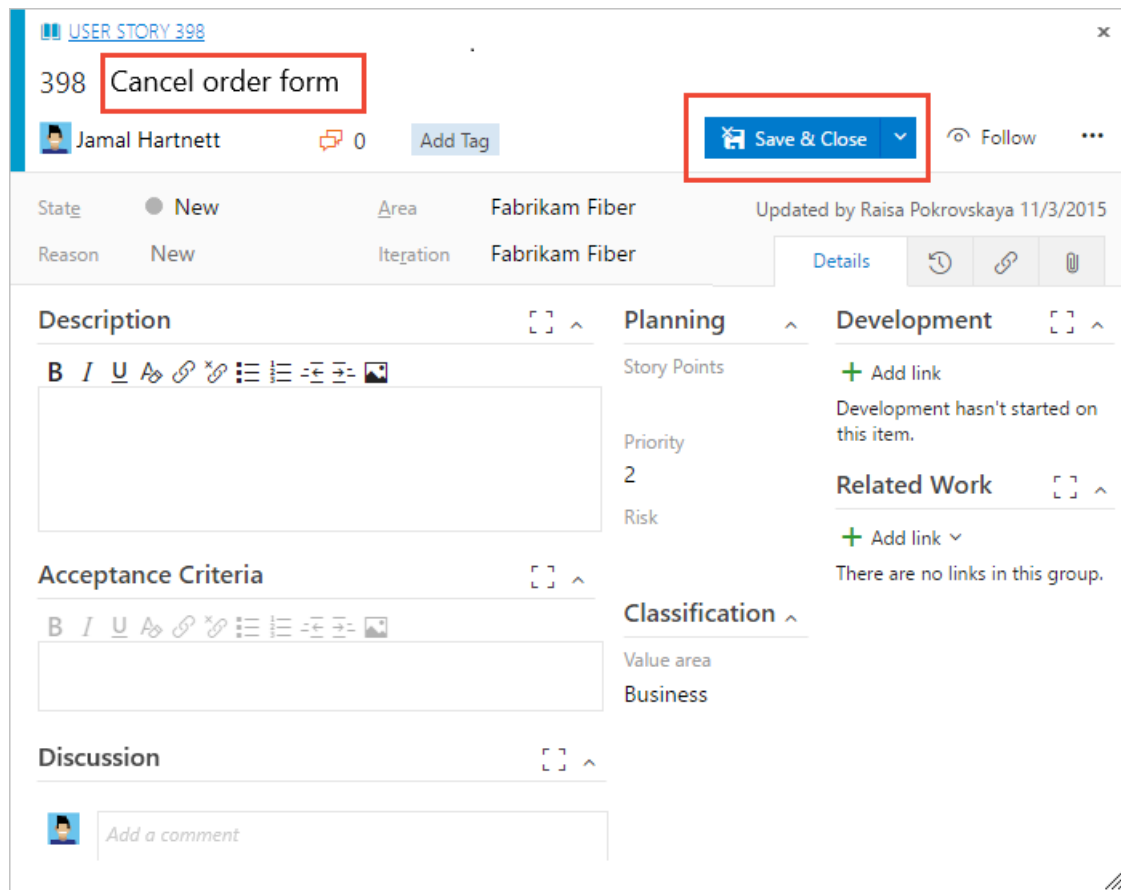
- **Assigned to me:** lists all work items assigned to you in the project in the order they were last updated. To open or update a work item, simply click its title.
- **Following:** lists work items that you've elected to follow.
- **Mentioned:** lists work items in which you've been mentioned in the last 30 days.
- **My activity:** lists work items that you have recently viewed or updated.
- **Recently updated:** lists work items recently updated in the project.
- **Recently completed:** lists work items completed or closed in the project.
- **Recently created:** lists work items created within the last 30 days in the project.

3. To add a work item, simply choose the work item type from the **New Work Item** drop down menu.

For example, here we choose User Story.



4. Enter a title and then save the work item. Before you can change the State from its initial default, you must save it.



NOTE

A caution icon on a tab indicates values that violate validation rules. You must correct information on that tab in order to save the work item.

New navigation isn't supported on TFS at this time. Choose **Previous navigation** for guidance.

For descriptions of each field, see [Work item field index](#).

You can [add existing tags to any work item to support filter backlogs and queries](#).

NOTE

Depending on the process chosen when the project was created, the types of work items you can create will differ. For example, backlog items may be called user stories ([Agile](#)), product backlog items ([Scrum](#)), or requirements ([CMMI](#)). All three are similar—they describe the customer value to deliver and the work to be performed. For an overview of all three processes, see [Choose a process](#).

Check the backlog, add work items to the backlog

Work appears in the backlog in priority order. Work item types may include bugs depending on the settings made for the team.

- [New navigation](#)
- [Previous navigation](#)



1. (1) Check that you have selected the right project, (2) choose **Boards>Backlogs**, and then (3) select the correct team from the team selector menu.

Order	Assigned To	State	Title
1	Jamal Hartnett	Committed	Hello World Web Site
2	Jamal Hartnett	Committed	Slow response on inform:
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 another team, open the selector and select a different team or choose the [Browse all sprints](#) option. Or, you can enter a keyword in the search box to filter the list of team backlogs for the project.

- Fabrikam Fiber Team
- Management team
- Phone
- Voice
- Web
- Browse all team backlogs

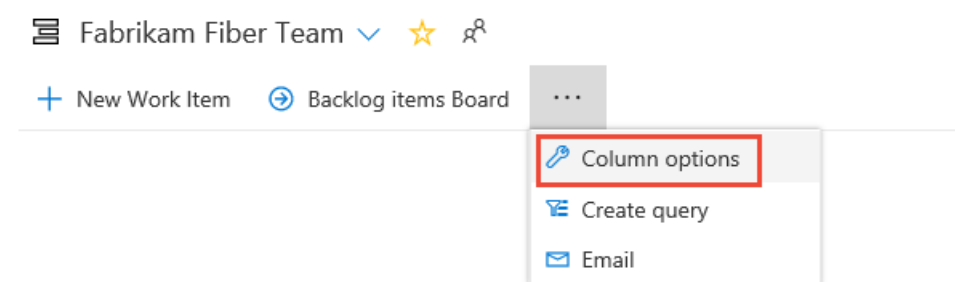
TIP

Choose the  star icon to favorite a team backlog. Favorited artifacts ( favorited icon) appear at the top of the team selector list.

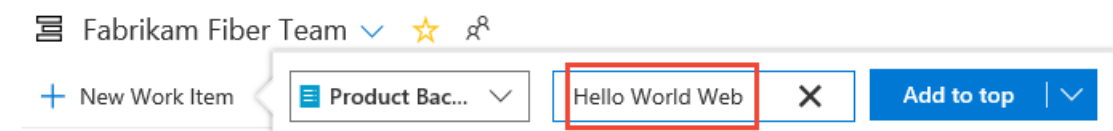
2. Check that you have selected **Backlog items** (for Scrum), **Stories** (for Agile), or **Requirements** (for CMMI) as the backlog level.



3. (Optional) To choose which columns should display and in what order, choose the **...** actions icon and select **Column options**. To learn more, see [Change column options](#).



4. To view or edit a work item, select it and choose **Enter**.
5. To add a work item, choose the **+ New Work Item**, enter a title and then press the Enter key or choose **Add to top**.



Repeat this step to capture all your ideas as work items.

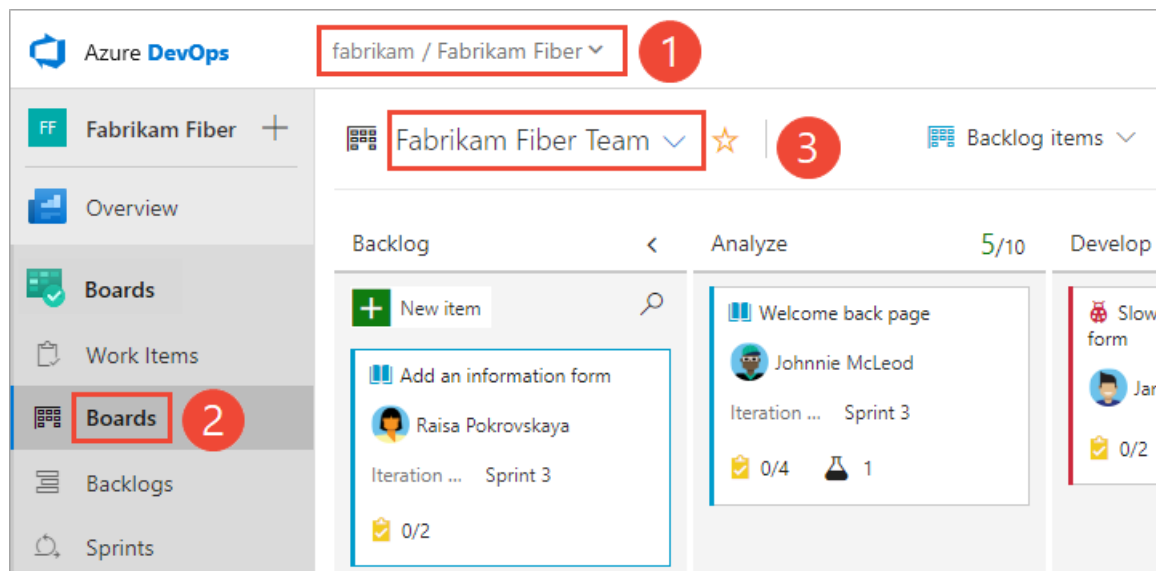
New navigation isn't supported on TFS at this time. Choose **Previous navigation** for guidance.

Check work in progress

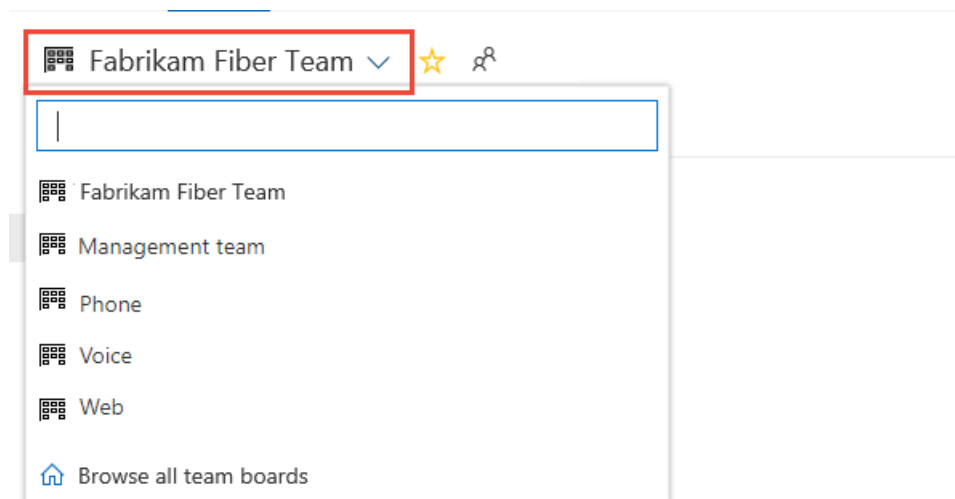
To view the team's progress, open the Kanban board. To view or edit a work item, choose a title and press **Enter**, or double-click the title.

- [New navigation](#)
- [Previous navigation](#)

1. (1) Check that you have selected the right project, (2) choose **Boards>Boards**, and then (3) select the correct team from the team selector menu.



To choose another team's board, open the selector and select a different team or choose 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

Choose the star icon to favorite a team board. Favorited artifacts (favored icon) appear at the top of the team selector list.

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

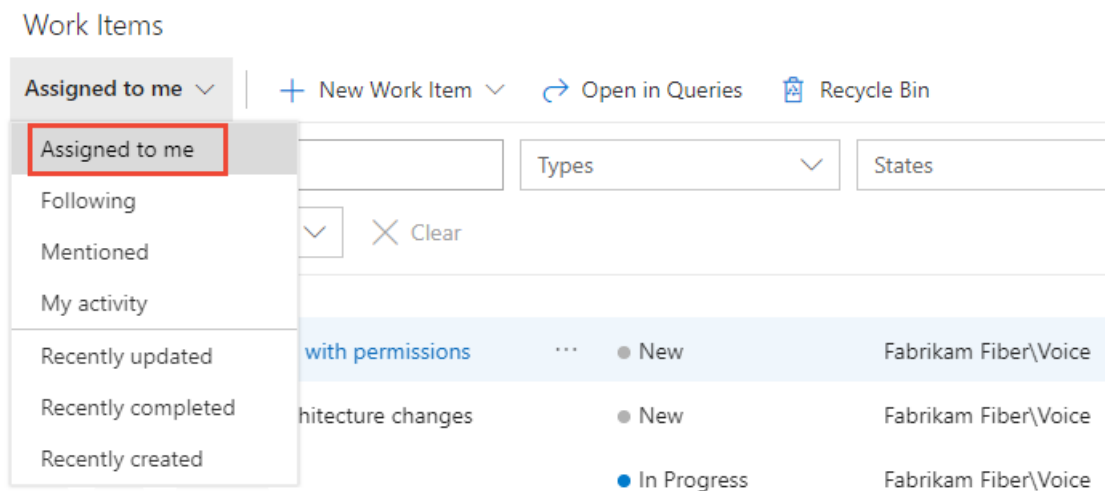


New navigation isn't supported on TFS at this time. Choose **Previous navigation** for guidance.

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

- [New navigation](#)
- [Previous navigation](#)

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



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](#).

New navigation isn't supported on TFS at this time. Choose **Previous navigation** for guidance.

Related articles

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

- [Add work items](#)
- [Create your backlog](#)
- [Kanban quickstart](#)
- [Provide Stakeholders access to edit build and release pipelines](#)

If you want to provide a group of users access to provide feedback, then you don't need to give them Stakeholder access. Instead, simply [give reviewers permissions to provide feedback](#).

Change individual or group permissions, grant select access to specific functions

9/10/2018 • 4 minutes to read • [Edit Online](#)

Azure DevOps Services | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

The standard way for permissions to accrue to individuals are by adding user accounts to one or more built-in security groups. However, in certain instances, you'll want to grant additional permissions to select individuals, but perhaps not all permissions assigned to the security group. For example, you might want to grant several individuals the ability to add or edit area and iteration paths, but not have all permissions available to members of the Project Administrators group.

The three ways to change permissions for an individual are:

- Create a custom group, define permissions for that group, add the user account to the group
- For object-level permissions: Add the user account and set permissions
- For project or collection-level permissions: Search for the user account and selectively change their permission assignments

In this topic you'll learn how to:

- Create a custom security group
- Set permissions for a custom security group
- Add members to a custom security group
- Change the permission assignments for an individual user account

If you are new to administering permissions and groups, review [About permissions and groups](#) to learn about permission states and inheritance.

NOTE

The images you see from your web portal may differ from the images you see in this topic. These differences result from updates made to Azure DevOps Services or your on-premises TFS. However, the basic functionality available to you remains the same unless explicitly mentioned.

Create a custom security group

Create a custom security group at the project-level or the collection-level. The method for creating a custom security group is the same, no matter at what level you add it.

To create a project-level security group, open the web portal and choose the project where you want to add users or groups.

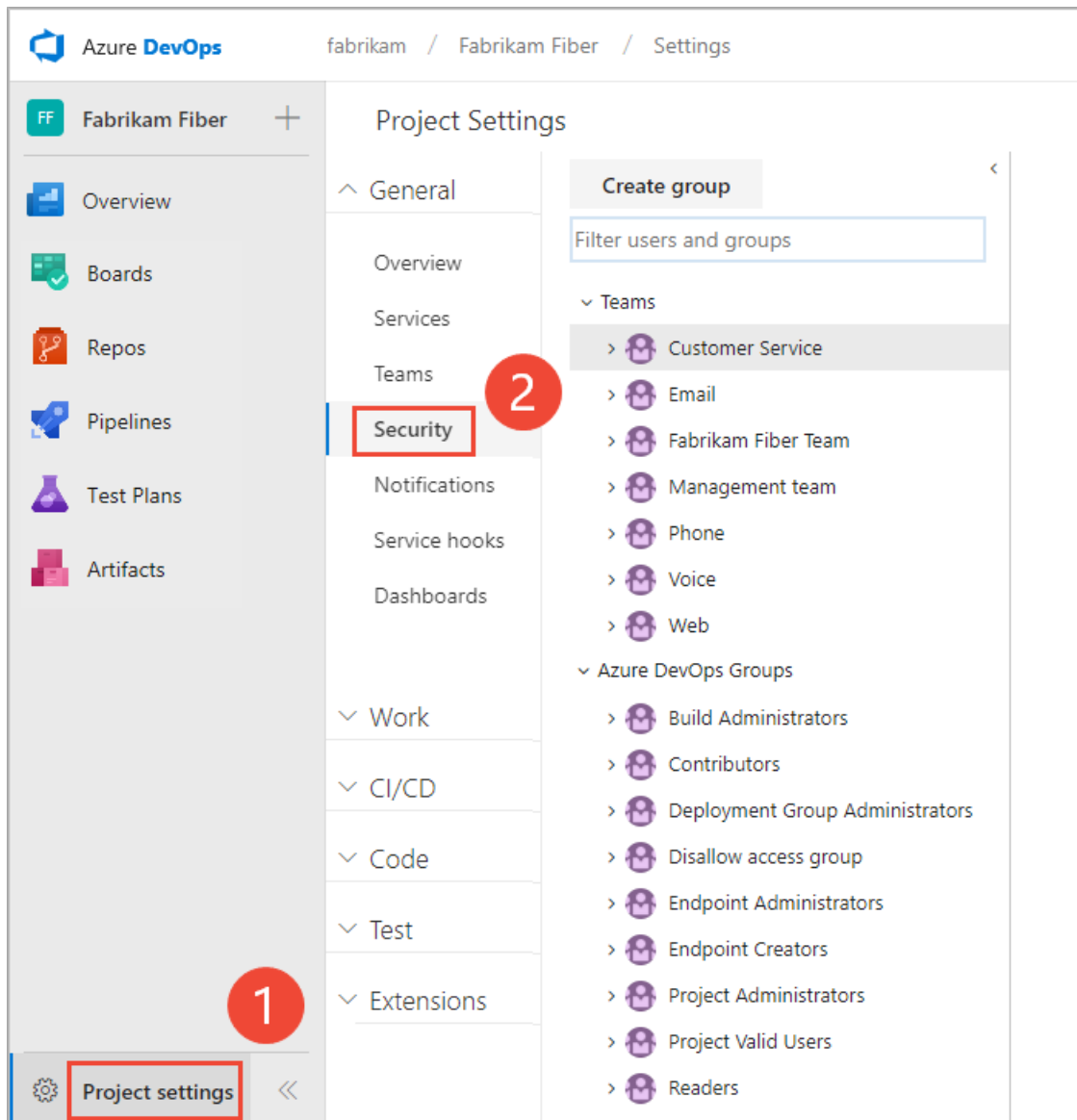
NOTE

Choose **Previous navigation** when you see a top-level blue bar. Choose **New navigation** if you see a vertical sidebar or if you enabled the **New Navigation** preview feature. The vertical sidebar, along with other navigational features, is enabled when the **New Navigation** preview feature has been enabled for the signed-in user or the organization. To learn how to use the web portal effectively, see [Web portal navigation](#).

For on-premises TFS, choose **Previous Navigation** for guidance.

- [New navigation](#)
- [Previous navigation](#)

1. Choose **Project Settings** > **Security**.



2. Choose **Create group** to open the dialog for adding a group.

Create group

Filter users and groups

Teams

Customer Service

Email

Fabrikam Fiber Team

Management team

Phone

Voice

Web

Azure DevOps Groups

Build Administrators

Contributors

Deployment Group Administrators

Disallow access group

Endpoint Administrators

Fabrikam Fiber > Customer Service | Edit...

Permissions

Members

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	Not set
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

3. Enter a name for the group, and optionally a description.

For example, here we define a Team Admins group.

Create new Azure DevOps Services group

PROFILE

Group name

Team admins

Description

Add all Team Admins to this group to provide them enhanced permissions

Create group

Cancel

4. Choose **Create group**.

New navigation isn't supported on TFS at this time. Choose **Previous navigation** for guidance.

Set permissions for a custom security group

1. To set permissions for the custom group you just created, choose the group name and then set one or more permissions.

Overview Work **Security** Version Control Policies Agent Queues Notifications

Create group <

Filter users and groups

> Teams

▼ Azure DevOps Groups

- > Build Administrators
- > Contributors
- > Disallow access group
- > Project Administrators
- > Project Valid Users
- > Readers
- > Release Administrators
- > **Team Admins**

Fabrikam Fiber > Team Admins | Edit... ▼

Permissions Members Member of

Bypass rules on work item updates	Allow
Create tag definition	Allow
Create test runs	Allow
Delete and restore work items	Allow
Delete team project	Deny
Delete test runs	Not set
Edit project-level information	Not set
Manage project properties	Deny
Manage test configurations	Allow
Manage test environments	Allow
Move work items out of this project	Allow
Permanently delete work items	Allow
Rename team project	Deny
Suppress notifications for work item updates	Not set
View analytics	Allow (inherited)
View project-level information	Allow
View test runs	Allow

[Clear explicit permissions](#)

Save changes Undo changes

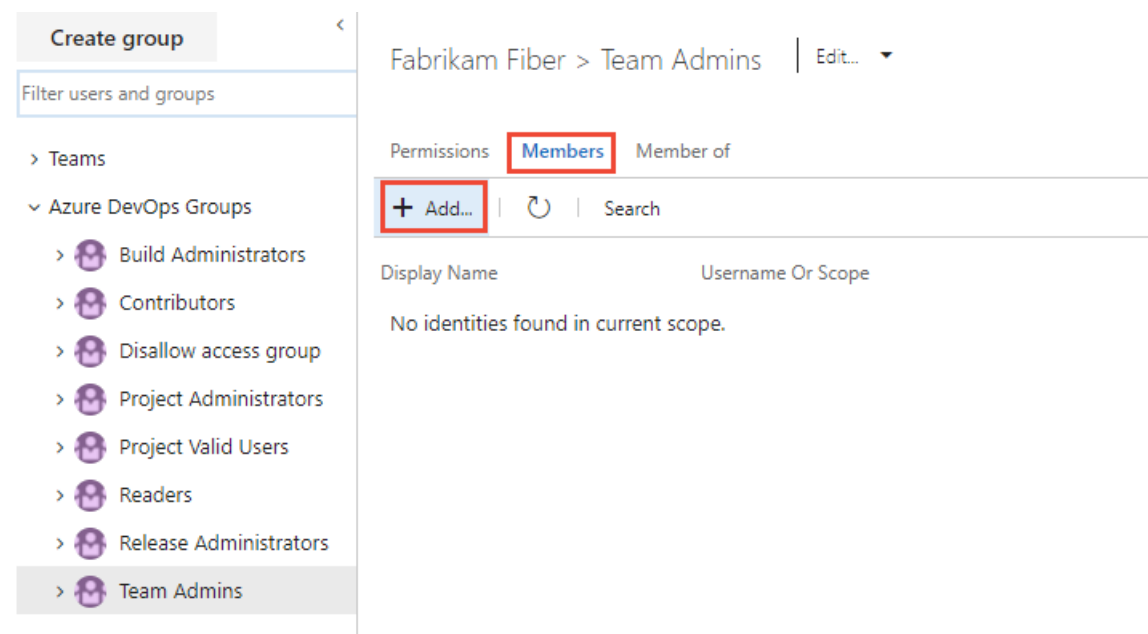
For a description of each permission, see [Permissions and groups reference, project-level permissions](#).

2. Choose **Save changes**.

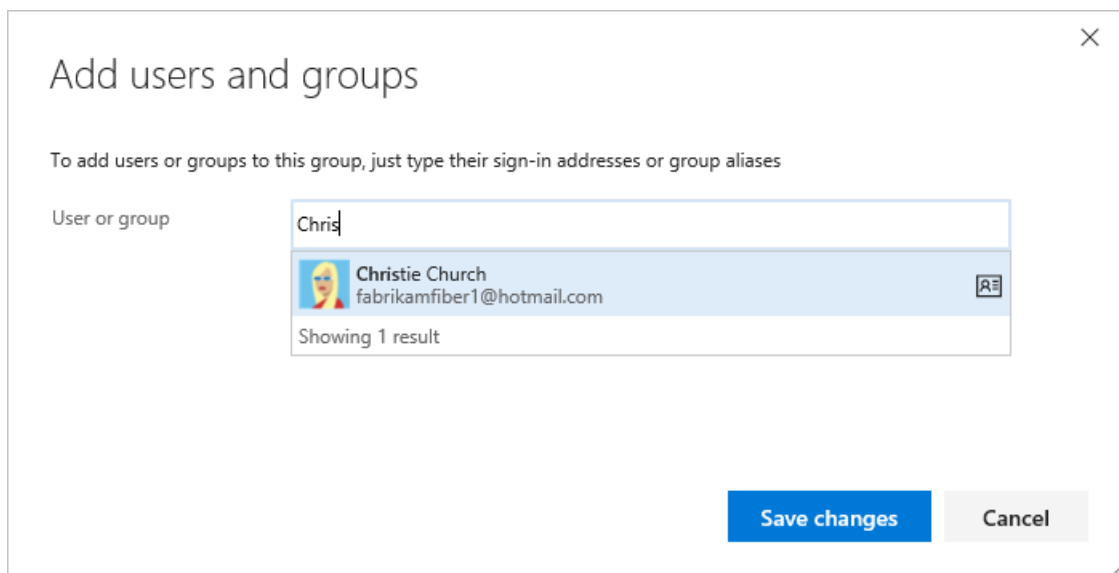
Add members to a custom security group

You add members to a custom security group in the same way you add users to a built-in group.

1. Choose the security group, choose **Members**, and then choose **Add**.



2. Type the name of the user account into the text box. You can type several identities into the text box, separated by commas. The system will automatically search for matches. Choose the match(es) that meets your choice.



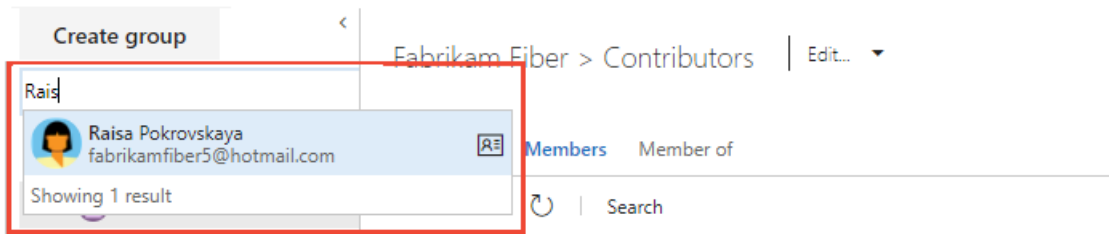
NOTE

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](#).

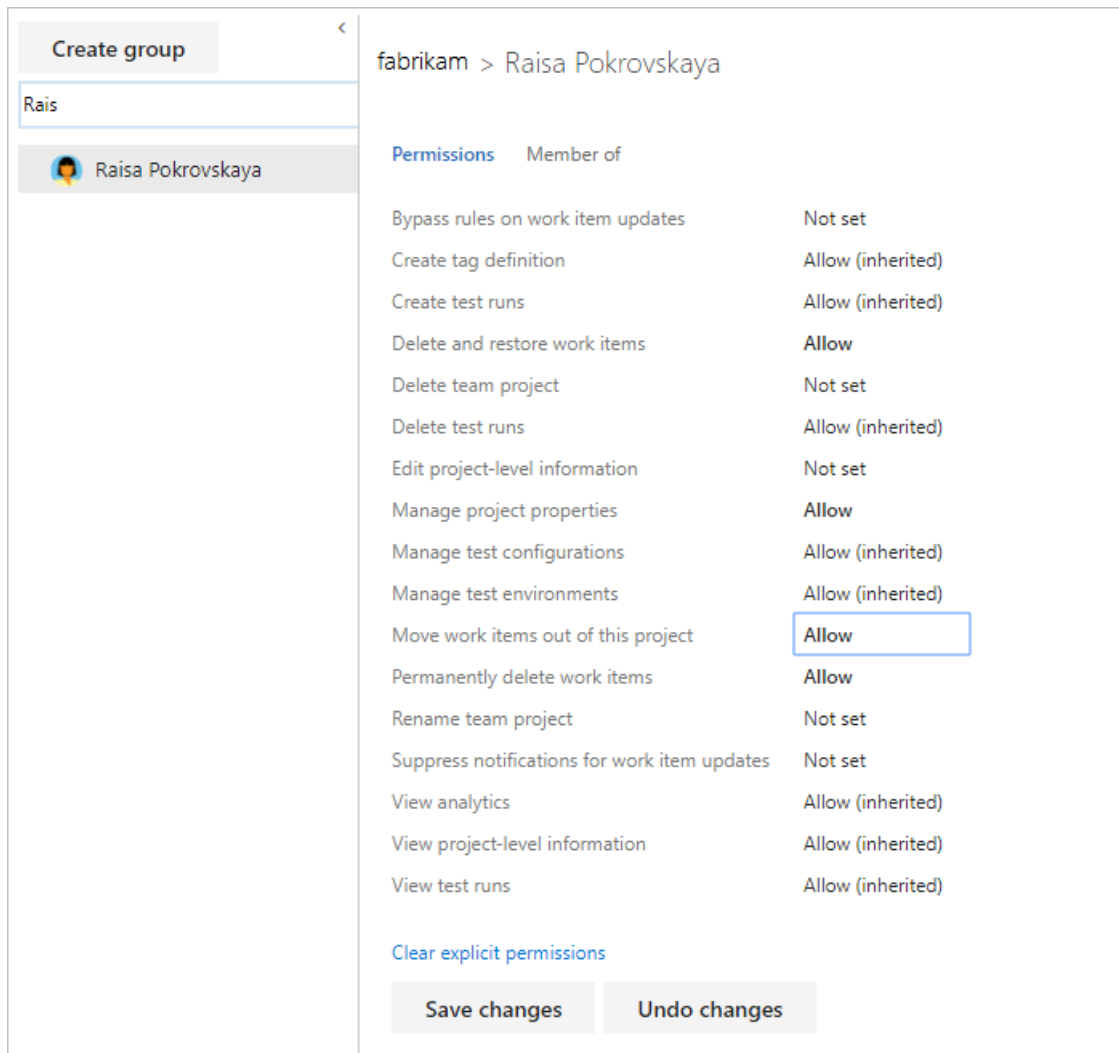
Change the permission assignments for an individual

To change the permission at a project-level

1. From the project-level **Security** page, type the name of the user account in the **Filter users and groups** box and select the account whose permissions you want to change.



2. Change the permissions for the account, setting a permission as **Allow** or **Deny**.



For a description of each permission, see [Permissions and groups reference, project-level permissions](#).

3. Choose **Save changes**.

To change the permission at a collection level

1. Open the account-level or collection-level **Security** admin page and follow the instructions provided in the previous section for project-level permissions.

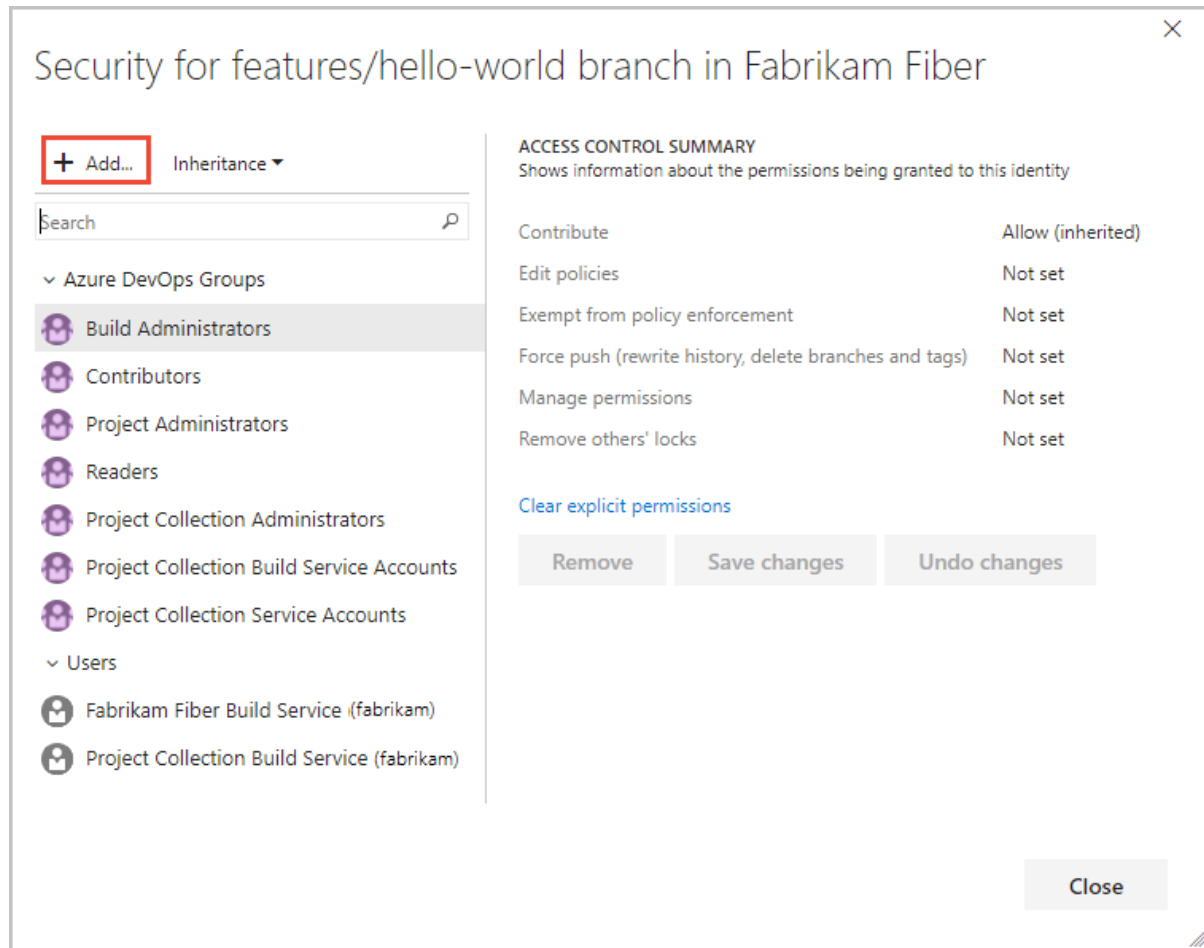
For a description of each collection-level permission, see [Permissions and groups reference, collection-level permissions](#).

To change the permission at an object-level

1. From the web portal, open the Security dialog for the object whose permissions you want to set. For specific instructions, see these topics:

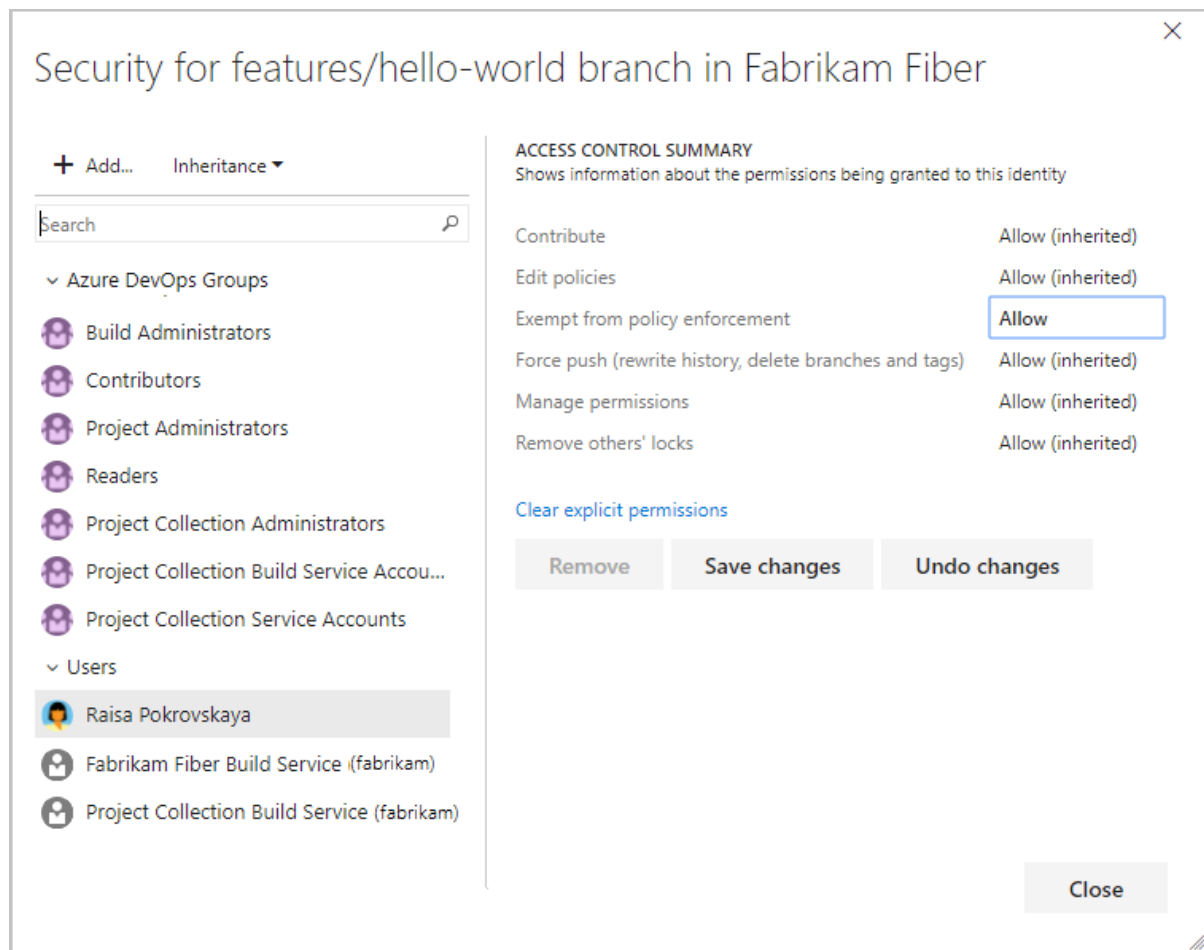
WIKI & DASHBOARD PERMISSIONS	DEVOPS PERMISSIONS	AGILE/WORK TRACKING PERMISSIONS
<ul style="list-style-type: none"> - README & Wiki - Dashboards 	<ul style="list-style-type: none"> - Git branch - Git repository - TFVC - Builds - Release pipeline security - Approvals and approvers 	<ul style="list-style-type: none"> - Area and iteration paths - Work item query and folder - Plan permissions

2. From the Security dialog, choose **Add** to add a user account.



3. Type the name of the user account, choose search, and select the account you want.

4. Select the user name from the left pane and then update the permission assignments, setting **Allow** or **Deny** for specific permissions.



For a description of a specific permission, see [Permissions and groups reference](#).

5. Choose **Save changes**.

Next steps

[Grant or restrict access to select features](#)

Related articles

- [About permissions and groups](#)
- [Permissions and groups reference](#)
- [Set permissions at the project-level or project collection-level](#)

Grant or restrict access to select features and functions

9/10/2018 • 6 minutes to read • [Edit Online](#)

Azure DevOps Services | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

You can grant or restrict access to resources that you manage in Azure DevOps Services or Team Foundation Server (TFS). Depending on your project needs, 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 are new to administering permissions and groups, review [About permissions and groups](#) to learn about permission states and inheritance.

Use this topic to learn:

- Recommended method for granting and restricting permissions
- How to delegate tasks by assigning select permissions to specific roles
- How to restrict access to view or modify objects
- How to 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 will determine 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. This 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 allows 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 which come with default permissions and role assignments are:

- Readers
- Contributors
- Team Administrator (role)
- Project Administrators
- Project Collection Administrators

For a summary of permissions provided to the above roles, see [Default permissions and access](#), or for the Project

Collection Administrators, see [Add administrators](#)

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 repository permissions for Git or TFVC .
Software architect (Git)	Manage repositories	Create repositories, Force push, and Manage permissions See Set repository permissions for Git or TFVC .
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 Add administrators, set permissions at the project-level or project collection-level .
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 Add administrators, set permissions at the project-level or project collection-level .
Process template manager (Hosted XML process model)	Work tracking customization	Edit collection-level information, See Add administrators, set permissions at the project-level or project collection-level .
Project management (On-premises XML process model)	Work tracking customization	Edit project-level information, See Add administrators, set permissions at the project-level or project collection-level .

Permissions manager	Manage permissions for a project, account, or collection	<p>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 grant permissions, See Add administrators, set permissions at the project-level or project collection-level.</p> <p>You can also grant permissions to manage permissions for the following objects:</p> <ul style="list-style-type: none"> • Manage Git or TFVC repository permissions • Manage Git branch permissions • Administer build and release permissions • Manage Wiki permissions.
---------------------	----------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Restrict access to view or modify objects

Azure DevOps and TFS are designed to enable all valid users to be able 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 [Change individual permissions, grant select access to specific functions](#).

Area to restrict	Permissions to set to Deny
View or contribute to a repository	View, Contribute See Set repository permissions for Git or TFVC .
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 based on a user or group

For the [Hosted XML process model](#) and [On-premises XML process model](#), you can customize work item types to support these restriction requests:

- Restrict who can create or modify a work item
- Restrict who can create specific work item types, such as Epics or Features

You achieve this by adding a rule to the work item type, usually within the **WORKFLOW** section. To learn more, see [Add a rule to a work item type, Apply or ignore rules based on user or group](#).

NOTE

These restriction types aren't available for Azure DevOps organizations and the [Inheritance process model](#).

Try this next

[Remove user accounts](#)

Related articles

- [Default permissions and access](#)
- [Permission lookup guide](#)
- [About permissions and groups](#)
- [Permissions and groups reference](#)
- [Set permissions at the project-level or project collection-level](#)

Key concepts for working with Azure DevOps Services and TFS

9/10/2018 • 6 minutes to read • [Edit Online](#)

Azure DevOps Services | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

The set of platforms, services, and tools you have access to through Azure DevOps Services can be overwhelming. Before you start using our products, you'll want to become familiar with how they fit together. You'll gain that understanding here as well as pointers to additional topics and tutorials to gain confidence in using our products to develop your software.

Collaborative, integrated software development

Azure DevOps Services, our hosted cloud offering, and Team Foundation Server (TFS), our on-premises platform, provide small teams as well as enterprises the services and tools to support developing and continuously deploying software. Even sole developers can use our platforms to manage their software and deploy their apps.

The three main areas that support software development include:

- Source control to manage versioning of software files
- Tracking tools to support planning and tracking work, code defects, issues and more
- DevOps tools to support building, testing, and continuous release of software apps.

Source control

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 (distributed) or Team Foundation Version Control (TFVC), a centralized, client-server system. Both systems enable you to check-in files and organize files within folders, branches, and repositories.

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 his or her own local repository, and 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 codebase. Later, they can merge, publish, or dispose of the branch.

NOTE

Git in Azure DevOps Services and TFS is standard Git. You can use Visual Studio with third-party Git services, and you can also use third-party Git clients with TFS.

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.

Work tracking and Agile tools

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 planned and track work using one or more tools such as 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.

Our systems provide several types of work items which 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 team members update as information becomes available and 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.

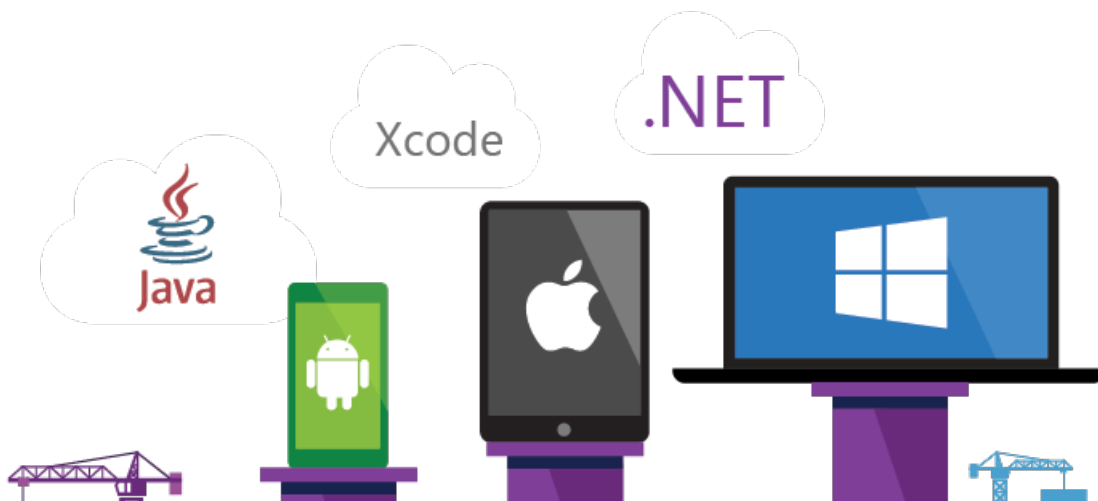
- Product backlog, used to create and prioritize stories or requirements
- Kanban, used to visualize and manage the flow of work as it moves from inception to in progress to done
- Sprint backlogs, used to plan work to complete during a sprint cycle, a regular 2 to 4 week cadence teams use when implementing Scrum
- Task board, used during daily Scrum meetings to review work completed, remaining, or blocked

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

DevOps and continuous integration

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



Scaling

Both Azure DevOps Services and TFS are enterprise-ready, supporting teams of any size, from tens to thousands. Azure DevOps Services provides a scalable, reliable, and globally available hosted service. It is backed by a 99.9% SLA, monitored by our 24x7 operations team, and available in local data centers around the world.

To learn more, see [About projects and scaling your organization](#). Also, for stories and short videos on how Microsoft transitioned from waterfall to Agile, see [Scaling Agile Across the Enterprise](#).

Customization and configuration of resources

You can configure and customize most services and applications to support your business needs or the way your teams work. For a comprehensive view of what 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 their 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 details, see [Customize an inheritance process](#).
 - **Build and release:** You can fully customize your build and release pipelines, define build steps, release environments, and deployment schedule. For details, see [Azure Pipelines](#).
 - **Test:** You can define and configure test plans, test suites, and test cases as well as configure test environments; additionally you can add test steps within your build pipelines. For details, see [Exploratory & Manual Testing](#), [Load testing](#), and [set up continuous testing for your builds](#).
-
- **Dashboards:** Each team can [configure their set of dashboards](#) to share information and monitor their 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 details, see [Customize the On-premises XML process model](#).
 - **Build and release:** You can fully customize your build and release pipelines, define build steps, release environments, and deployment schedule. For details, see [Azure Pipelines](#).
 - **Test:** You can define and configure test plans, test suites, and test cases as well as configure test environments; additionally you can add test steps within your build pipelines. For details, see [Exploratory & Manual Testing](#), [Load testing](#), and [set up continuous testing for your builds](#).

Extensibility

In addition to all the pre-built functionality available to you, you can add to it in the following ways:

- [Visual Studio Marketplace](#) : Provides extensions that you can install either on your organization, server, or Visual Studio client
- [Service hooks](#): Enable you to perform tasks on other services when events happen within your project hosted on Azure DevOps Services or TFS
- [REST APIs](#): Provide the ability to create custom extensions that plug into Azure DevOps Services or TFS
- [Visual Studio SDK](#): Helps you extend Visual Studio features or integrate new features into Visual Studio. You can distribute your extensions to other users, as well as to the Visual Studio Marketplace.

Resources

- [Pricing](#)

Define your Azure DevOps organizations and projects

9/10/2018 • 5 minutes to read • [Edit Online](#)

Azure DevOps Services | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

In Azure DevOps, an organization is a mechanism for organizing and connecting groups of related projects. Examples are business divisions, regional divisions, or other organizational structure. You can choose one organization for your entire company, or separate organizations for specific business units, or an organization just for you.

For a larger company, you may want to create multiple organizations using different user accounts (most likely Azure Active Directory accounts). Think about what groups and users in your company share strategies and work, and group them into specific organizations. For example, the (fictional) Fabrikam company might create three Azure DevOps organizations: Fabrikam-Marketing, Fabrikam-Engineering, and Fabrikam-Sales. Each organization will have a separate URL, such as <https://dev.azure.com/Fabrikam-Marketing>, <https://dev.azure.com/Fabrikam-Engineering>, and <https://dev.azure.com/Fabrikam-Sales>. The organizations are all for the same company but are mostly isolated from each other.

Choose your organization admin account type

You can create one or more Azure DevOps organizations. These organizations can be created by using a Microsoft account or with an Azure Active Directory (Azure AD)-backed account. This account provides the credentials to sign in to your new Azure DevOps organization at `https://dev.azure.com/{yourorganization}`.

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.

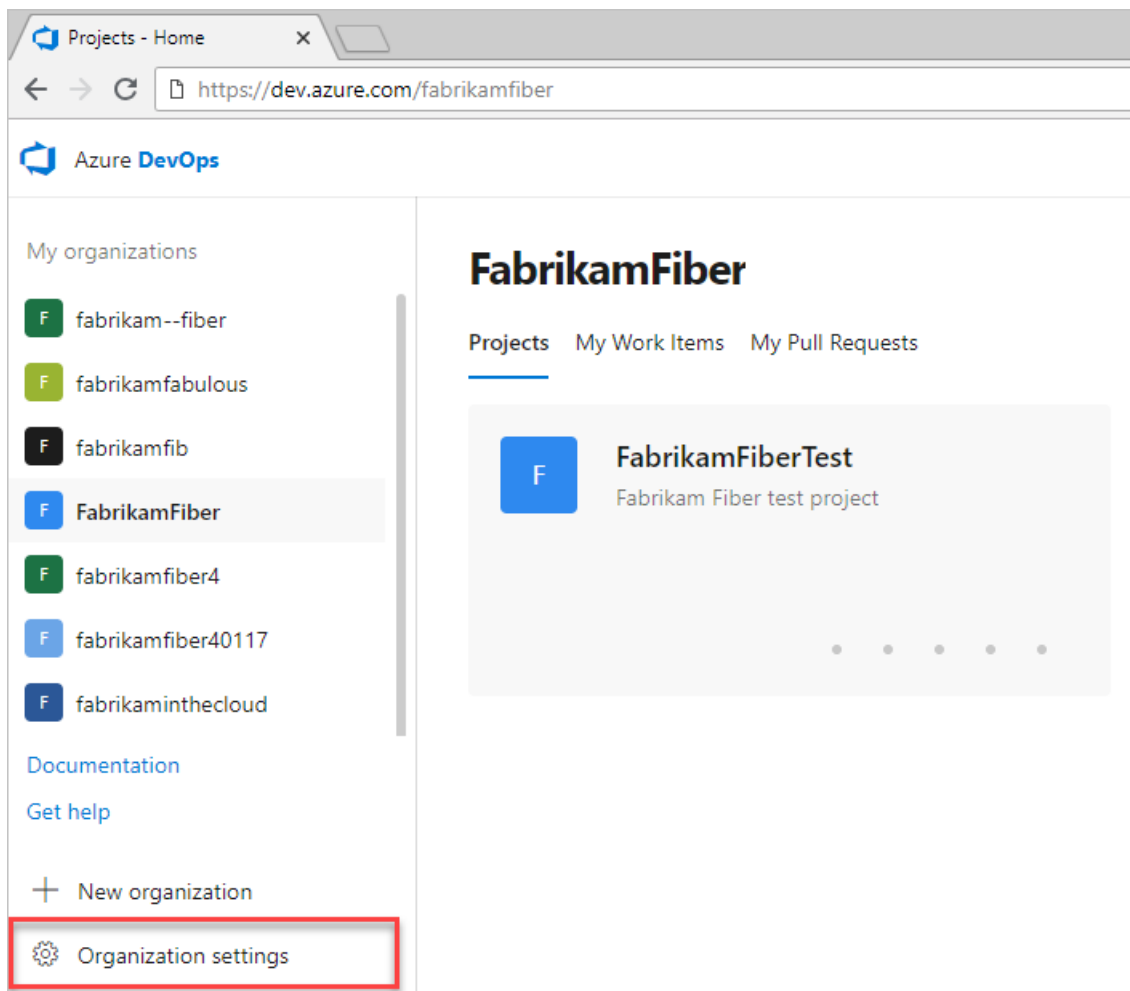
If you don't have an Azure Active Directory instance, you can either [create one for free](#) from the Azure portal or use your Microsoft account to create an organization. An example is `johndoe@outlook.com`.

Azure Active Directory-backed account

Use your work or school account managed by its Azure Active Directory instance. If you use Azure or Office 365, you might have one already. If you don't, learn how to [sign up for Azure Active Directory](#) to **automatically connect** your Azure DevOps organization to your Azure Active Directory. All users must be members in that directory to access your organization. To add users from other organizations, use [Azure AD B2B collaboration capabilities](#).

Define organizations

Organization settings are managed by administrators. As the creator of the organization, you're an administrator by default. You can access those settings by using the **Organization settings** button in the lower-left of your Azure DevOps portal.



For more information on configuring an organization, read [Create an organization](#).

Define projects

Each organization contains one or more projects. Each project contains a 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, and continuous test integration throughout the life cycle.

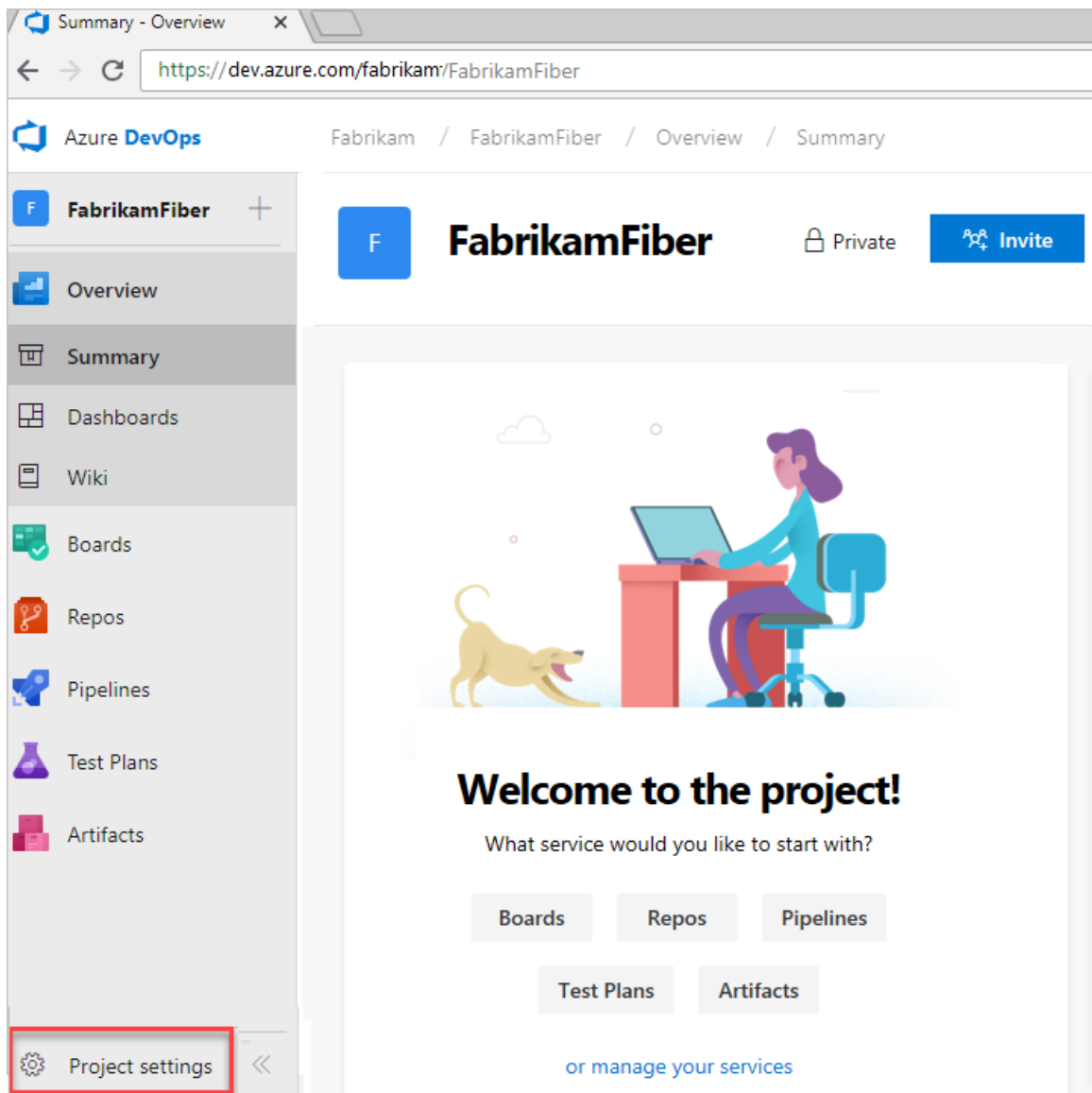
Within an organization, you can have one large single project or multiple projects. Choose either of the following:

- Create a single project that contains many repos and teams.
- Create multiple projects, each containing its own set of teams, repos, builds, work items, and other elements.

Projects can be created or removed as you need. Think about the specific strategic work scoped to one of the organizations you created previously and who should have access to it. Use this information to name and create a project. This project will have 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 by visiting its URL and selecting the **Project settings** button at the lower-right of the page.



For more information on configuring a project, read [Create a project](#).

Single project

You might have a large product or service that's managed by many teams. Those teams have tight interdependencies on each other across the product life cycle. You create a project and divide the work by 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.

[!RECOMMENDATION]

When multiple teams work on the same product, we recommend that you have all teams on the same iteration schedule. This arrangement helps keep your teams aligned and delivering value on the same cadence.

A high volume of queries and boards can make it difficult 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. To help alleviate this issue, make sure that you use good naming conventions and a simple folder structure. When you add a new repo to your project, it's a good time to reflect on your strategy and determine if that repo can be placed into its own project.

Multiple projects

Most companies work on several products or services at a time. In those cases, we recommend having multiple projects. A project is best determined 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 multiple projects creates some alignment challenges. 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.

[!CONSIDER THE FOLLOWING]

- Use the same process across all your projects.
- Enforce the same iteration schedules across all projects.

Azure DevOps provides cross-project experiences when it comes to managing work. You can easily create cross-project queries and move work items from one project to another.

If the projects stored in multiple repos work on independent schedules or processes, then splitting them into multiple projects might make the most sense. When you're considering multiple projects, note that Git repo portability makes it easy to move a repo between projects and still retain full-fidelity commit history. Other history cannot be migrated between projects. Examples are push and pull request history.

Try this next

[Create an organization](#) or [Create a project](#)

Or, after you've created a new organization and project in Azure DevOps, you can begin sharing your code with others: [Code with git](#).

Source control

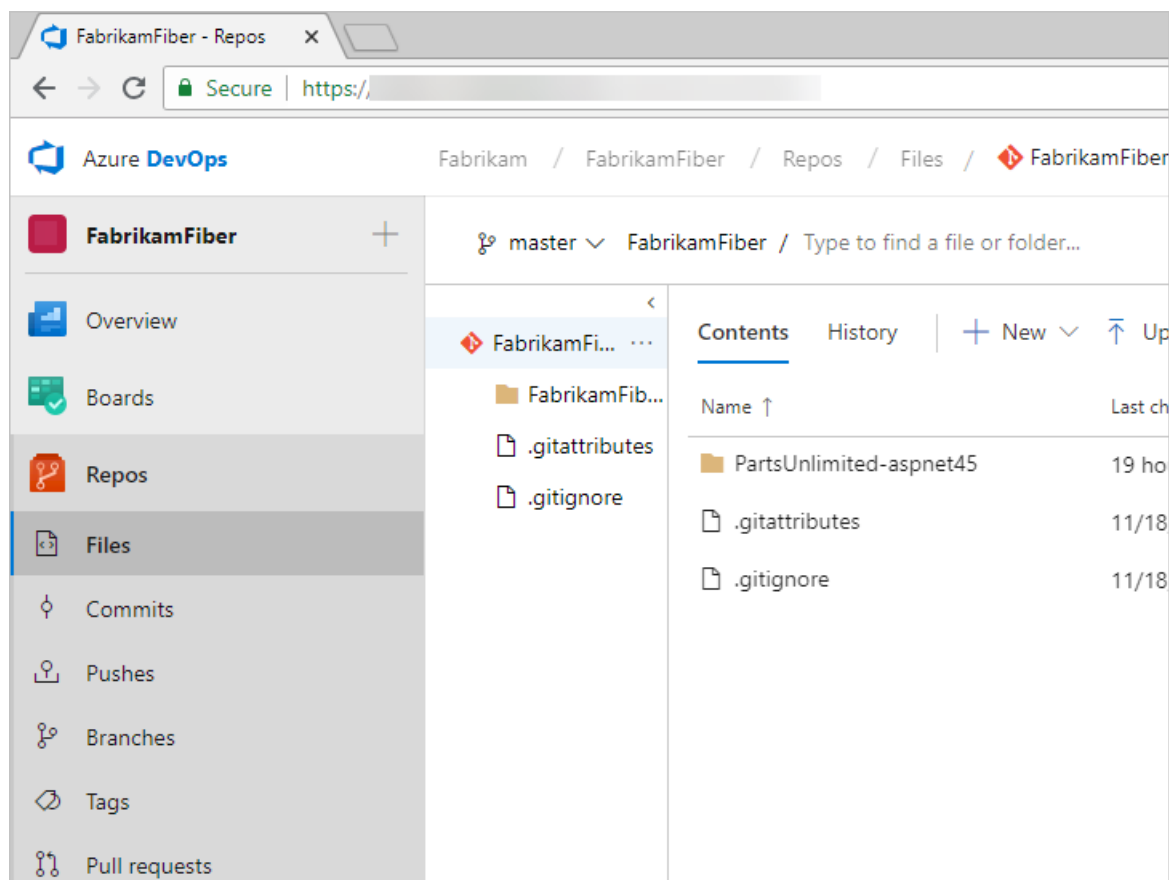
9/10/2018 • 2 minutes to read • [Edit Online](#)

Azure DevOps Services | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

A source control system (also called a *version control* system) allows 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 (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.

You manage your repos, branches, and other code development operations from **Azure Repos**.



With Git, each developer has a copy of the source repository, including all branch and history information, on their dev machine. Each developer works directly with their own 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, Azure DevOps Services, and TFS is standard Git. You can use Visual Studio with third-party Git services. You can also use third-party Git clients with TFS.

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.

Try this next

Start sharing your code or get your code by using source control.

[Code with Git](#)

What is Azure Boards?

9/10/2018 • 6 minutes to read • [Edit Online](#)

Azure Boards | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

Azure Boards provides a suite of interactive Agile tools with which you can plan and track work, bugs, and issues. Azure Boards is available from Azure DevOps Services and Team Foundation Server (TFS).

Agile, a term coined in 2001 in the [Agile Manifesto](#), encompasses approaches to software development that emphasize incremental delivery, team collaboration, continual planning, and continual learning. The set of Agile tools that Azure Boards provides are designed to support teams working with Agile methodologies, such as Kanban and Scrum. To learn more, see [What is Agile?](#).

All tools support viewing and defining work items. Each work item represents an object stored in the work item data store. Each work item is assigned a unique identifier, an ID, which is unique within an account or project collection.

Your Agile tool set, available from **Azure Boards**, consists of the following main interactive lists and signboards. Each of these pages provide a filtered set of work items.

NOTE

The **New navigation** feature provides a vertical navigation experience and is in preview for Azure DevOps Services. When you [enable new navigation](#), you automatically enable several new Agile tool features that are described in the [New Work Hubs](#) blog post.

On-premises Microsoft Team Foundation Server users can select **Previous navigation** for guidance.

- [New navigation](#)
- [Previous navigation](#)
- **Work items:** Use to quickly find work items assigned to you or pivot or filter work items based on other criteria
- **Boards:** Use to implement Kanban practices and visualize the flow of work for a team
- **Backlogs:** Use to plan, prioritize, and organize the work for a team to do within a product or portfolio backlogs
- **Sprints:** Use to plan work for a team to perform during a specific time frame referred to as a sprint
- **Queries:** Use to define a set of filter criteria to list work items for the purposes of sharing with others or performing bulk updates
- **Plans:** Use to review the schedule of stories or features your teams plan to deliver. Plans show scheduled work items defined assigned to sprints (iteration path) of selected teams against a calendar view.

New navigation isn't supported on TFS at this time. Choose **Previous navigation** for guidance.

Teams and Agile tools

A team refers to a group of project members that work in a particular product area. Those areas are represented as **area paths**, hierarchical paths denoting the possible areas of ownership in an organization. A team is defined by a name, its members, and its area paths.

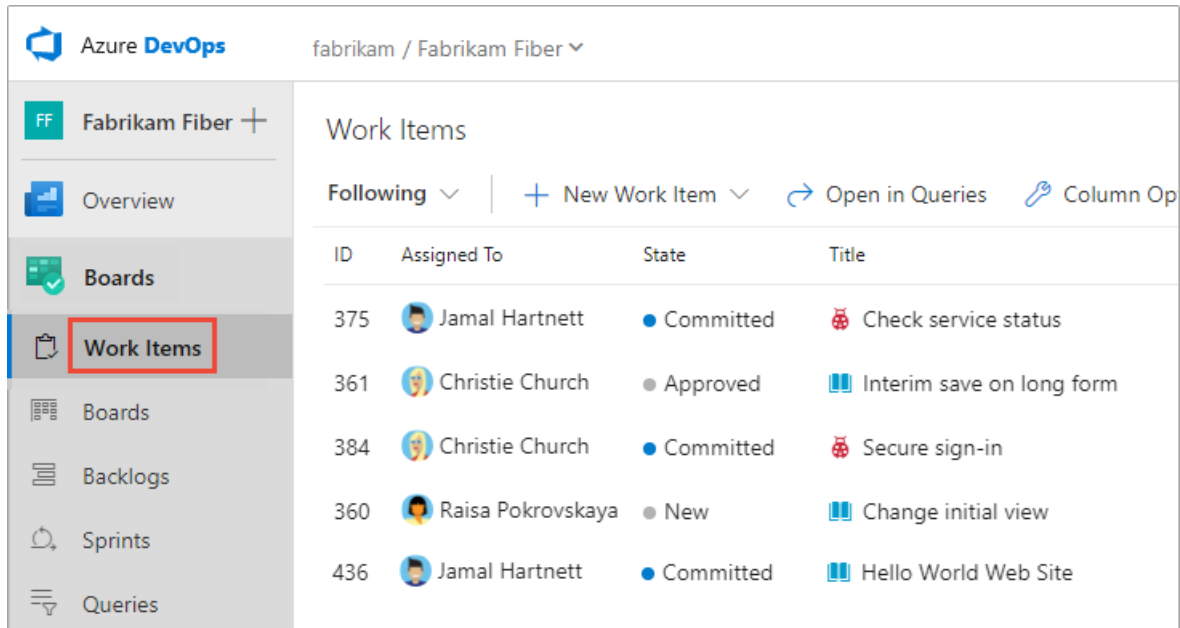
These Agile tool—**Boards**, **Backlogs**, **Sprints**, and **Plans**—rely on team configurations. For example, if you want to add a Kanban board or product backlog, you define a team. For more information on teams, see [About teams and Agile tools](#).

Your view and options available will differ somewhat depending on if you have [enabled the New Navigation feature](#), which displays a vertical navigation interface along with several changes to navigation of Agile tools.

Work Items and work item types

Open **Work Items** to access several personalized pivots and filter functions to focus on work items you care about. You can quickly find work items assigned to you, that you're following, or have viewed or modified recently—even when defined for different teams and projects. To learn more, see [View and add work items](#).

- [New navigation](#)
- [Previous navigation](#)



The screenshot shows the Azure DevOps interface for the 'fabrikam / Fabrikam Fiber' project. The left sidebar contains a vertical navigation menu with the following items: 'Fabrikam Fiber' (with a plus icon), 'Overview', 'Boards', 'Work Items' (highlighted with a red box), 'Backlogs', 'Sprints', and 'Queries'. The main content area is titled 'Work Items' and shows a table of work items. The table has columns for ID, Assigned To, State, and Title. The work items listed are:

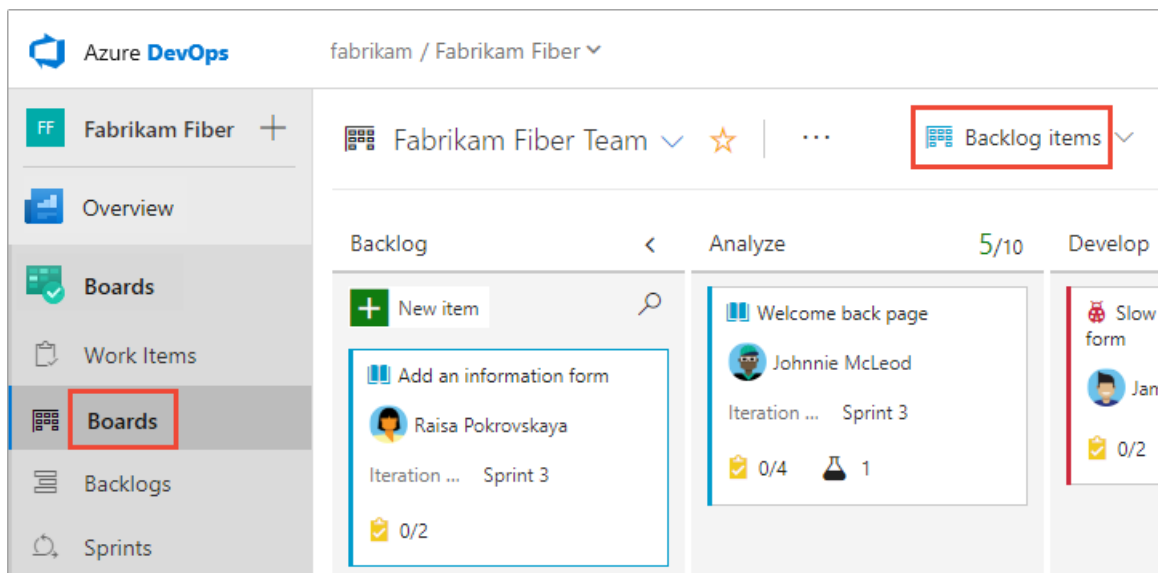
ID	Assigned To	State	Title
375	Jamal Hartnett	Committed	Check service status
361	Christie Church	Approved	Interim save on long form
384	Christie Church	Committed	Secure sign-in
360	Raisa Pokrovskaya	New	Change initial view
436	Jamal Hartnett	Committed	Hello World Web Site

New navigation isn't supported on TFS at this time. Choose **Previous navigation** for guidance.

Boards

Boards present work items as cards and support quick status updates through drag-and-drop, similar to sticky notes on a physical whiteboard. Each board supports many Kanban practices such as defining columns and swimlanes, setting Work-in-Progress (WIP) limits, defining the Definition of Done, and more. To get started, see [Kanban quickstart](#).

- [New navigation](#)
- [Previous navigation](#)

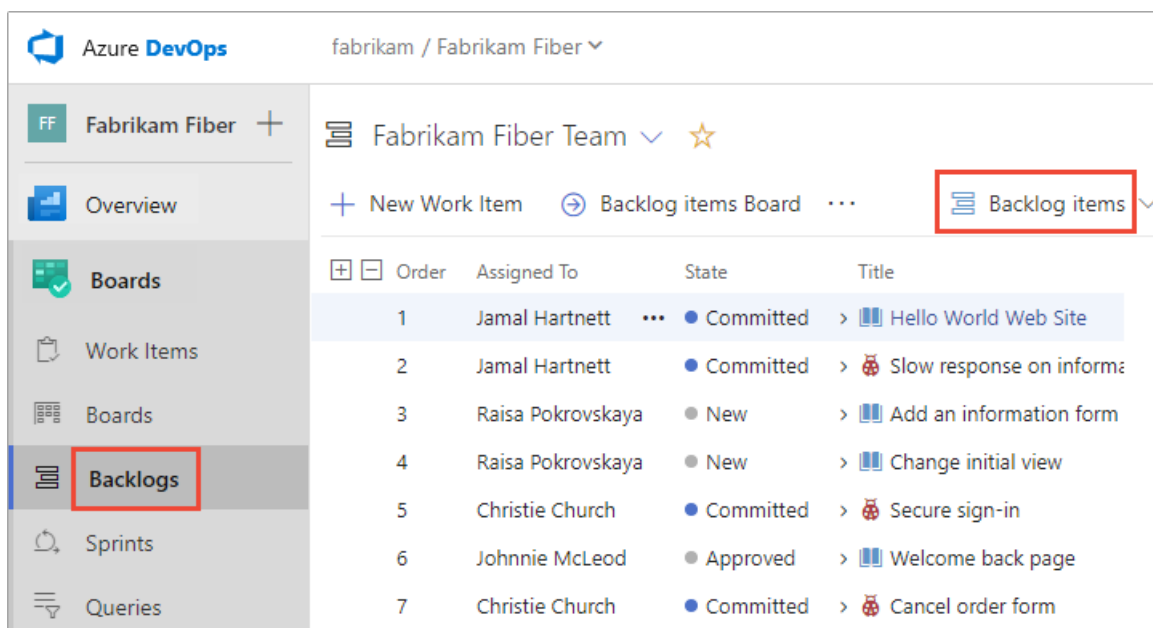


New navigation isn't supported on TFS at this time. Choose **Previous navigation** for guidance.

Backlogs

Backlogs present work items as lists. A product backlog represents your project plan, the roadmap for what your team plans to deliver. Your backlog also provides 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. To get started, see [Create your backlog](#).

- [New navigation](#)
- [Previous navigation](#)



New navigation isn't supported on TFS at this time. Choose **Previous navigation** for guidance.

Sprints

Sprint backlogs and taskboards provide a filtered view of work items a team has assigned to a specific iteration path, or sprint. Sprints are defined for a project and then selected by teams. From your backlog, you can map work to an iteration path using drag-and-drop, and then view that work in a separate **sprint backlog**.

- [New navigation](#)
- [Previous navigation](#)

Azure DevOps fabrikam / Fabrikam Fiber

Fabrikam Fiber Team June 11 - June 29 7 work days remaining

Taskboard Backlog Capacity + New Work Item ... Sprint 1

	Order	Assigned To	Remaining Work	Title
+	1	Jamal Hartnett	...	> Hello World Web Site
	2	Raisa Pokrovskaya	6	> Cancel order form
	3	Jamal Hartnett	5	> GSP locator interface
	4	Jamal Hartnett	3	> Request support
	5	Jamal Hartnett		Check service status
	6	Raisa Pokrovskaya	8	> Cancel order form
	7	Raisa Pokrovskaya	14	> Phone sign in

New navigation isn't supported on TFS at this time. Choose **Previous navigation** for guidance.

You can also filter the cards on your taskboard to show only those cards mapped to a given sprint. It is recommended that an entire organization share the same sprint interval in order to align multiple teams in a single org to the same rhythm. A common sprint rhythm is sometimes referred to as the "heartbeat" of an org.

Queries

Queries are filtered lists of work items based on criteria that you define using a query editor. You use queries to find groups of work items with something in common, to triage a set of items to prioritize or assign them, or to create status and trend charts that you can then add to dashboards. To get started, see [Create a managed query](#).

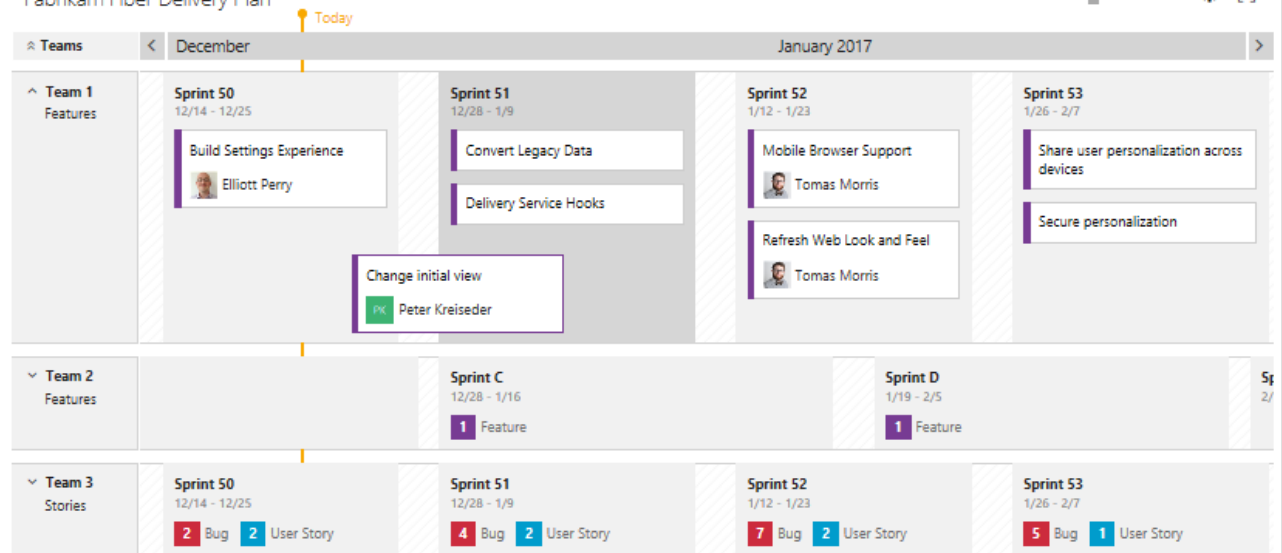
All Queries > My Queries > Active bugs 8 work items 1 selected

Results Editor Charts Run query + New Save query 4 of 8

ID	Assigned To	State	Title	Area Path
390	Jamal Hartnett	Committed	Cancel order form	Fabrikam Fiber\Voice
375	Jamal Hartnett	Committed	Check service status	Fabrikam Fiber\Phone
516	Christie Church	New	Performance issues	Fabrikam Fiber\Phone
487	Raisa Pokrovskaya	New	Check issues with permissions	Fabrikam Fiber\Voice
377	Christie Church	Approved	Switch context issues	Fabrikam Fiber\Phone
384	Christie Church	Committed	Secure sign-in	Fabrikam Fiber\Phone
364	Jamal Hartnett	Committed	Slow response on information form	Fabrikam Fiber\Web
400	Johnnie McLeod	Approved	Canadian addresses don't display ...	Fabrikam Fiber\Web

Delivery plans

Delivery plans display work items as cards along a timeline or calendar view. This can be an effective communication tool with managers, partners and stakeholders for a team or for several teams collaborating on specific features or requirements.



Tools and clients that connect to Azure DevOps Services and TFS

9/10/2018 • 8 minutes to read • [Edit Online](#)

Azure DevOps Services | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

Our platform of software development tools began over 20 years ago with the releases of Visual Basic and Visual Studio as an integrated development environment (IDE). Visual Studio supports a number of plug-ins which extend its functionality. In particular, the Team Explorer plug-in allows the Visual Studio client to connect to Azure DevOps Services and TFS to support source control, work tracking, build, and test operations.

The set of tools available to you that interface with Azure DevOps Services and TFS include the following:

- Desktop client developer tools
- Office integration tools
- Web based tools
- Command-line tools
- Marketplace extensions
- REST APIs

Desktop client developer tools

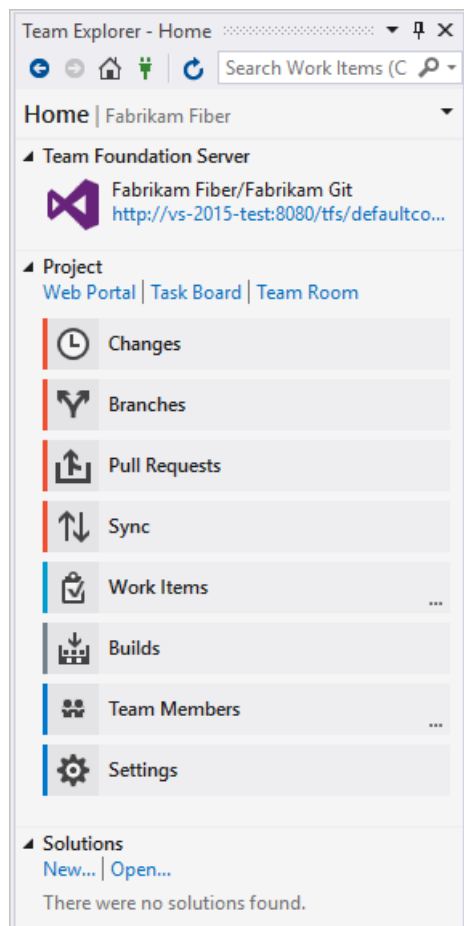
Developers have access to a variety of tools through these versions of Visual Studio and plug-ins. To download any version of Visual Studio, visit the [Visual Studio Downloads page](#). To understand what features you get with the Visual Studio versions, see [Compare Visual Studio Offerings](#).

- **Visual Studio Community:** A fully-featured and extensible IDE for creating modern applications for Android, iOS, and Windows, as well as 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 those with a need to scale Supports designing, building and managing complex enterprise applications
- **Visual Studio Test Professional:** Provides access to Microsoft Test in addition to development tools to support quality and collaboration throughout the development process
- **Visual Studio Team Explorer:** A free solution for non-developers to interact with Team Foundation Server and Visual Studio Team Services.
- **Eclipse/Team Explorer Everywhere:** Free plug-in to support teams running Eclipse on Linux, macOS, or Windows that connect to Azure DevOps Services or TFS
- **Android Studio with the Azure DevOps Services Plugin for Android Studio:** Free plug-in to support Android developers and connect to Git repositories on Azure DevOps Services or TFS
- **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 Services or TFS
- **Visual Studio Code:** Free, open-source code editor with a free extension to support connecting to Git repositories on Azure DevOps Services or TFS.

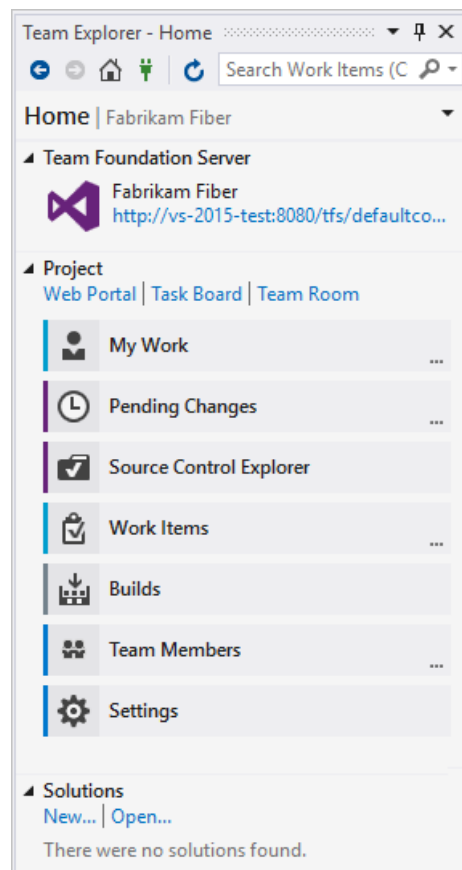
Team Explorer plug-in

Team Explorer, a plug-in to all Visual Studio versions, provides connects Visual Studio to projects defined in Azure DevOps Services or TFS. You can manage source code, work items, and builds. To learn more, see [Work in Team Explorer](#).

HOME PAGE WITH GIT



HOME PAGE WITH TFVC



Office integration tools

When you install any edition of Visual Studio or [Team Foundation Server Standalone Office Integration 2015 \(free\)](#), the Team Foundation plug-in installed to integrate 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:** 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.
- **Project Professional:** With Project Professional and the Team Foundation Server Extensions for Project Server you can manage projects that synchronize data that exists in both TFS and Project Server. Project managers and software development teams can use the tools that they prefer, work at the level of precision that supports their needs, and easily share information.
- **PowerPoint Storyboarding:** Lets you illustrate user stories and requirements using PowerPoint. The Team Foundation plug-in installs to your existing PowerPoint client.

IMPORTANT

Support for integrating TFS with Project Server is deprecated for TFS 2017. However, synchronization support is provided by a third party. See [Synchronize TFS with Project Server](#) for details.

Task-specific clients

The following clients support specific tasks, such as managing testing efforts, providing feedback, or modifying work items.

- **Microsoft Test Manager:** Allows you to manage your test efforts, create and run manual tests, and create and track bugs that are found during test efforts. Test Manager installs with Visual Studio Test Professional and Visual Studio Enterprise.
- **Test & Feedback extension (previously called the Exploratory Testing extension):** Provides a lightweight, plug-in to a web browser. Stakeholders can respond to feedback requests for user stories and features generated in Azure DevOps Services or TFS. This extension is free to stakeholders.
- **Microsoft Feedback Client:** Your stakeholders can use **Microsoft Feedback Client** to record feedback for your application as video, audio, or type-written comments. This client installs with all versions of Visual Studio, or can be [installed from the free download](#). All feedback is stored in the work item data store and requires [stakeholders to have required permissions](#).

IMPORTANT

Test Manager is deprecated for TFS 2017.

Browser-based web tools

Web portal

The collaboration tools supported through the web portal are summarized under [Essential services](#). On Azure DevOps Services, new features are deployed usually every three weeks, and for TFS, usually quarterly. For release notes, see [Azure DevOps Services Features Timeline](#).

You can use these browsers to access the web portal (Azure DevOps Services and TFS).

VERSION	EDGE	INTERNET EXPLORER	SAFARI (MAC)	FIREFOX	CHROME
Azure DevOps Services	most recent	11 and later	9.1 and later	most recent	most recent
TFS 15	most recent	11 and later	9.1 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

Edge, Firefox, and Chrome automatically update themselves, so Azure DevOps Services and TFS support the most recent version.

To learn more, see [Web portal navigation](#).

Browser-based extensions

The following extensions are available from the Visual Studio Marketplace and are built and maintained by the Azure DevOps Services product team.

- **Test Manager:** Run tests using your browser with simple pass/fail of steps, add comments/attachments, take screenshots and file bugs. You can accomplish all of this with automatic end-to-end traceability.
- **Package Management:** Build packages of reusable code components and share them. The Package Management extension enables continuous delivery workflows by supporting multiple packaging protocols such as NuGet and npm . It makes packages available to your team, your builds, and your releases.
- **Code search:** Increases cross-team collaboration and code sharing by enabling developers to quickly locate relevant information within the code base of all projects hosted within an organization or collection. With it, you can discover implementation examples, browsing definitions, and find error text.

- [Work item search](#): Quickly find relevant work items by searching across all work item fields over all projects in an organization. Perform full text searches across all fields to efficiently locate relevant work items. Use in-line search filters, on any work item field, to quickly narrow down to a list of work items.

Find additional extensions from the [Marketplace](#)

Application monitoring tools

To monitor your applications you can use Application Insights for web apps or HockeyApp for mobile apps.

Monitor web applications with Application Insights

Application Insights is an extensible Application Performance Management (APM) service for web developers. Use it to monitor your live web application. It will automatically detect performance anomalies. It includes powerful analytics tools to help you diagnose issues and to understand what users actually do with your app. It's designed to help you continuously improve performance and usability. It works for apps on a wide variety of platforms including .NET, Node.js and J2EE, hosted on-premises or in the cloud.

With Application Insights you can do the following:

- Gain actionable insights through application performance management and instant analytics
- Detect and diagnose exceptions and application performance issues
- Monitor Azure websites, including those hosted in containers, plus websites on-premises and with other cloud providers
- Seamlessly integrate with your DevOps pipeline using Azure DevOps Services, GitHub, and our webhooks
- Get started from within Visual Studio, or monitor existing apps without re-deploying

To learn more, see [Microsoft Azure - Application Insights](#).

Monitor mobile applications with HockeyApp

With HockeyApp you can develop, distribute, and beta-test your mobile apps. HockeyApp supports the following:

- Android, Cordova, iOS, macOS, Unity, Windows, and Xamarin apps
- Live, reliable crash reports
- Collect in-app feedback from real users
- Open-source SDKs to let you know what code is running in your apps
- Integration with your existing build system and work item management solution

To learn more, see [Microsoft Azure - HockeyApp](#).

Command-line tools

You can perform many code development and administrative tasks using command line tools, such as the following:

- [Git commands](#)
- [TFVC commands](#)
- [TFSConfig](#)
- [TFSDelProject](#)
- [TFSSecurity](#)
- [TFSServiceControl](#)
- [witadmin \(work item tracking\)](#)

Marketplace extensions

While Visual Studio, Azure DevOps Services, and TFS 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. Written with standard technologies—HTML, JavaScript, CSS—you can develop your own extensions using your preferred dev tools.

You build extensions using our RESTful API Library. You publish them to the Visual Studio Marketplace, where you can privately maintain them or share with millions of developers that use Visual Studio, Azure DevOps Services, and TFS.

To learn more, visit the [Marketplace](#) and [Overview of extensions](#).

REST APIs

The Azure DevOps Services and TFS 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 Services and TFS. To learn more, see [REST API overview](#).

Related articles

- [Key concepts](#)
- [Essential services](#)
- [Software development roles](#)
- [Pricing](#)

Software development roles supported by Azure DevOps Services and TFS

9/10/2018 • 4 minutes to read • [Edit Online](#)

Azure DevOps Services | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

If you are a sole developer, or work on a small team, chances are that you participate in many activities—performing tasks associated with issue tracking, feature planning, coding, testing, build, and deployment.

However, if you work within a large organization, you're probably more focused on a specific set of tasks that are traditionally aligned with one or two specific roles, such as software development, project management, and DevOps.

This topic provides a roadmap to support you in quickly coming up to speed on the features and tasks available to you based on the role you perform.

Contributor roles

Team members are contributors who have access to the code base, work item tracking, Agile tools, build pipelines, test tools, and more. If you need to lock down specific areas to a select set of contributors, you can do that through the [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 TFVC repository hosted in Azure DevOps Services or TFS. From the web portal or supported IDE, they can view repositories, check history, and more.

- To get started using Git, see one of these resources:
 - [Share your code with Git and Visual Studio](#).
 - [Share your code in Git using Eclipses](#).
 - [Share your code in Git using Xcode](#).
 - [Share your code in Git using IntelliJ](#).
 - [Get Started with Git and Azure DevOps Services](#).
- To get started using TFVC, see one of the following resources:
 - [Develop and share your code in TFVC using Visual Studio](#)
 - [Share your code in TFVC using Eclipse](#)
 - [Share your code in TFVC using Xcode](#)

Project managers

The role of project manager typically encompasses planning the feature set to deliver, setting priorities, and tracking the status of work, code defects, and customer issues. The suite of web-based Agile tools provide PMs with the views and features they need to perform these tasks. All work is 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

To get started, see [About Azure Boards and Agile tools](#).

If you are used to using Excel or Project to plan and track your work, you can still use these tools and integrate with Azure DevOps Services and TFS. See [Bulk modify using Excel](#) and [Create your backlog and tasks using Project](#).

DevOps: builders, testers, and release managers

One of the main advantages to working with Azure DevOps Services or TFS is the suite of tools and integrated functionality that supports build, testing, and deploying software applications. Here are the main DevOps associated tasks supported by Azure DevOps Services and TFS:

- Define builds
- Unit test your code
- Run tests with your builds
- Performance test your apps
- 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 provided here: [Azure Pipelines](#) and [Azure Test Plans](#).

Stakeholders

With stakeholder access, anyone within 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. They can 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 features 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. That said, there are still several administrative tasks required to support a collaborative, integrated software development environment

The main tasks are grouped as follows by membership within a security group or role:

Team administrators

Responsible for configuring team settings which include:

- Backlog and board settings
- Team area(s) and iterations (sprints)
- Team members
- Team dashboards
- Team work item templates
- Team alerts

To get started, see [Configure team settings](#).

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](#)
- [Customize work tracking objects](#)
- [Build agents, pools, and service connections](#)
- [Test](#) and [release](#) retention policies

Project collection administrators

Organization owners and project collection administrators

Responsible for configuring organization-level resources. These include:

- Manage billing
- Add and manage projects
- Manage collection-level permissions
- Customize work tracking processes
- Install and manage extensions (install custom or [Marketplace extensions](#))

To get started, see [Manage organizations](#) and [Settings](#).

Project collection administrators

Responsible for configuring collection-level resources. These include:

- Add and manage projects
- Manage collection-level permissions
- Install and manage extensions (install custom or [Marketplace extensions](#))

To get started, see [Settings](#).

TFS administrators

Responsible for installing, upgrading, and maintaining an on-premises TFS deployment. Tasks include:

- Install TFS
- Update servers running TFS
- Manage database backups
- Server administrative settings and permissions
- Build retention policies
- Add and manage project collections

To get started, see [Server Administration \(TFS\)](#).

Related articles

- [Key concepts](#)
- [Essential services](#)

Understand differences between Azure DevOps Services and TFS

9/10/2018 • 10 minutes to read • [Edit Online](#)

Azure DevOps Services | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

Azure DevOps Services and Microsoft Team Foundation Server (TFS) both provide an integrated, collaborative environment that supports Git, continuous integration, and Agile tools for planning and tracking work.

Azure DevOps Services is the cloud offering that provides a scalable, reliable, and globally available hosted service. It is backed by a 99.9% SLA, monitored by our 24/7 operations team, and available in local datacenters around the world.

TFS is the on-premises offering that's built on a SQL Server back end. Enterprises usually choose on-premises TFS when they need their data to stay within their network or when they want access to SharePoint sites and SQL Server reporting services that integrate with TFS data and tools.

Although both offerings provide the same [essential services](#), compared with TFS, Azure DevOps Services gives organizations 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 organizational needs, consider the fundamental differences and differences in supported features between the two.

Fundamental differences between Azure DevOps Services and TFS

Consider the differences in these areas when contemplating a move from TFS to Azure DevOps Services:

- [Scope and scale data](#)
- [Authentication](#)
- [Users and groups](#)
- [Manage user access](#)
- [Security and data protection](#)

Differences in specific feature areas between Azure DevOps Services and TFS

Although Azure DevOps Services is a hosted version of TFS, there are some differences between the features that are available in the two products. Some TFS features are not supported in Azure DevOps Services at all. For example, Azure DevOps Services does not support integration with SharePoint or Project Server (which are now deprecated features for TFS 2018 and later versions).

Two additional areas that differ in their support are:

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

If you're on TFS and considering moving to Azure DevOps Services, read [Migrate data from TFS to Azure DevOps Services](#) to understand your options.

Scope and scale data

TFS scales using deployments, project collections, and projects

TFS has three options for scoping and scaling data: deployments, project collections, and projects. In the simplest case, deployments are just servers.

Deployments can also be more complicated, however, including everything from a two-server deployment, where SQL is split out on a separate machine, to high-availability farms comprising lots of servers.

Project collections serve as containers for security and administration, in addition to serving as physical database boundaries. They are 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.

Learn more: [Manage project collections](#).

Azure DevOps Services scales using organizations and projects

Azure DevOps Services is slightly different. It currently has only 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 contain exactly one project collection. Organizations can contain multiple projects, such as TFS project collections.

To prepare to use the organization entity, we recommend that you create organizations in Azure DevOps Services wherever you would have created collections in TFS. In the short term, having your work split across multiple organizations can cause some problems, but we plan to address these problems when the organization entity is introduced. In particular:

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

Learn more: [Define your Azure DevOps Services organizations and projects](#).

Authentication

With TFS, 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 usually transparent, and you never see any kind of sign-in experience.

With Azure DevOps Services, you connect over the public internet (for example, `https://contoso.visualstudio.com`). You'll either authenticate with [Microsoft Organization](#) 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 organizations configure their Azure DevOps Services organizations to use Azure AD rather than Microsoft Accounts. This provides a better experience in many scenarios and more options for enhanced security.

Learn more: [Access Azure DevOps Services with Azure Active Directory](#).

Users and groups

In TFS, you provide users access to deployments by adding AD groups to various TFS groups (for example, the Project 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 TFS.

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

Manage user access

In both Azure DevOps Services and TFS, 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 additional charge. You need to pay only for other users who need access.

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

Users with a TFS Client Access License (CAL) can have Basic access. Visual Studio subscribers can have either Basic or Advanced access, depending on their subscriptions. TFS does not attempt to verify these licenses or enforce compliance.

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 to more users, you'll need to set up billing for your organization and [pay for more users](#). Otherwise, all other users get Stakeholder access.

If you use Azure AD groups to provide access to groups of users, Azure DevOps Services will assign the appropriate access levels to them automatically when they sign in for the first time. For Azure DevOps Services organizations that are configured to use Microsoft Accounts for sign-in, you will have to assign access levels to each user explicitly.

Security and data protection

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

Process customization

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

- For Azure DevOps Services, you use the **Inheritance** process model, which supports WYSIWYG customization.
- For TFS, you use the **On-premises XML** process model, which supports customization through import or export of XML definition files for work-tracking objects.

Although the **On-premises XML** process model option is powerful, it also can cause a number of problems. Chief among them is that processes for existing projects do not update automatically when TFS is upgraded.

For example, TFS 2013 introduced several new features that depended on new work-item types and other process template changes. When you upgrade from TFS 2012 to TFS 2013, each project collection gets new versions of each of the "in the box" process templates that include these changes. However, these changes are not automatically incorporated in existing projects. Instead, after you finish upgrading, you have to include them in each project by using the [Configure Features](#) wizard or a more manual process.

To help you avoid these issues in Azure DevOps Services, custom process templates and **witadmin.exe** have always been disabled. This approach has enabled us to automatically update all projects with each Azure DevOps Services upgrade. Meanwhile, the product team has been 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 these new Azure DevOps Services process-customization capabilities, you can make customizations directly within the Azure DevOps Services web UI. If you want to customize your processes programmatically, you can do so through REST endpoints. When you customize projects in this way, those projects will continue to be updated automatically when we release new versions of their base processes with Azure DevOps Services upgrades.

To learn more, see [Customize your work-tracking experience](#).

Reporting

Both TFS and Azure DevOps Services have a variety of tools to give you insight into the progress and quality of your software projects. The tools include:

- [Dashboards](#) and lightweight [charts](#), available in both TFS and Azure DevOps Services. These tools are easy to set up and use.

In addition, Azure DevOps Services gives you access to the following services:

- [The Analytics service](#) and [Analytics widgets](#). The Analytics service is optimized for fast read-access and server-based aggregations.
- [PowerBI integration](#), which supports getting Analytics data into PowerBI reports, providing 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. Enterprise organizations can generate queries that span multiple projects or their entire Azure DevOps organization.

We plan to make these services available in TFS in a future release. See [Reporting roadmap](#).

The following reports and dashboards are available only in TFS:

- [Excel reports](#)
- [SQL Server Reporting Services \(SSRS\) reports](#)
- [SharePoint dashboards](#)

Related articles

- [Key concepts](#)
- [Essential services](#)
- [Client-server tools](#)
- [Software development roles](#)
- [Pricing - Azure DevOps Services](#)
- [Pricing - TFS](#)

Project management and navigation glossary

9/10/2018 • 6 minutes to read • [Edit Online](#)

Azure DevOps Services | TFS 2018 | TFS 2017

This glossary describes terms used when navigating in the web portal for Azure DevOps Services and Team Foundation Server (TFS). See also:

- [Agile glossary](#)
- [Security glossary](#)

Backlogs

An interactive list of work items that corresponds to a team's project plan or roadmap for what the team plans to deliver. The product backlog supports prioritizing work, forecasting work by sprints, and quickly linking work to portfolio backlog items. You can define your backlog items and then manage their status using the Kanban board.

Each product backlog can be customized by a team. Learn more: [Create your backlog](#).

Analytics views

Analytics views provide a simplified way to specify the filter criteria for a Power BI report based on the Analytics service. The Analytics service is the reporting platform for Azure DevOps Services.

Area paths

Area paths allow you to group work items by team, product, or feature area. Whereas, iteration paths allow you to group work into sprints, milestones, or other event-specific or time-related period. The area path allows you to define a hierarchy of paths. Learn more: [About area and iteration paths](#).

Boards (Kanban)

An interactive, electronic sign board that supports visualization of the flow of work from concept to completion and lean methods. Learn more: [Kanban basics](#).

Dashboards

User-configurable interactive signboards that provide real-time information. Dashboards are associated with a team and display configurable widgets to display information. Learn more: [Add and manage dashboards](#).

Extensions

Extensions are simple add-ons that are used to customize and extend the DevOps experience of Azure DevOps Services and TFS. They are written with standard technologies—HTML, JavaScript, CSS—and can be developed using your preferred development tools. Hundreds of extensions are available from the [Visual Studio Marketplace](#).

Favorites

A method for tagging an object to support quick navigation by yourself or other team members. You can tag work item queries and build definitions as personal and team favorites. Other objects you can favorite for yourself only include code branches, delivery plans, test plans, and teams or projects. Learn more: [Set personal or team favorites](#).

Follow

A tool for tagging specific work items or pull requests for which you want to receive email updates when changes are made to them. Learn more: [Follow a work item or pull request](#).

Git repository

A Git repository supports a distributed version control system for tracking changes, reviewing contributions to the code, and more. Each developer has a copy of the source repository on their dev machine. You can add multiple Git repositories to a project. Learn more: [Git Repositories](#).

NOTE

Git in Visual Studio and Azure DevOps Services is standard Git. You can use Visual Studio with third-party Git services, and you can also use third-party Git clients with Azure DevOps Services.

Notifications

With notifications, you receive an email when changes occur to work items, code reviews, pull requests, source control files, and builds. For example, you can get notified whenever a bug that you opened is resolved, or when a work item is assigned to you. You receive notifications based on rules or subscriptions made by you, for your teams, or for the project. Learn more: [About notifications](#).

Pipelines

Pipelines are artifacts that you define to run concurrent builds or deploy concurrent releases. Two types of pipelines are supported, private and hosted. To learn more, see [CI/CD concurrent jobs](#).



Plans (aka delivery plans)

A configurable view that displays work from multiple teams and projects laid out within a calendar based on each team's iterations. Each row in the view represents the work from a team's product or portfolio backlog, with each card corresponding to a work item—user story, feature, or epic. Learn more: [Review team delivery plans](#).

Process

Defines the building blocks of the work tracking system. To customize a process, you first create an inherited process from one of the default system processes—[Agile](#), [Scrum](#), or [CMMI](#). Changes you make to a process are seen by all projects that use it. Learn more: [About process customization and inherited processes](#).

Projects

A project (previously referred to as a *team project*) provides a repository for source code and a place for a group of people to plan, track progress, and collaborate on building software solutions. A project is defined for an Azure DevOps Services organization or within a TFS project collection. It provides support for focusing on those objects defined within the project. Learn more: [About projects and scaling your organization](#).

Public projects

A project created within an Azure DevOps Services organization that is visible to the whole world. Everyone in the world can discover them and perform limited operations. Administrators can control who gets to fully contribute.

Administrators can switch a project from private to public, and vice-versa, as described in [Change the project visibility](#).

Queries

A query supports finding and listing work items. Queries are used to support managed searches which you can use to triage work versus adhoc searches used to find a specific work item. Learn more: [About managed queries](#).

Repositories

A source control folder or container you configure to help you track file changes in. You can have any number of repository on your computer, each stored in their own folder. Each repository is independent, so changes saved in one repository don't affect the contents of another. Learn more: [Create a new Git repo](#).

Sprints (aka Iterations)

A time period, usually two to three weeks, used to group work items to be completed during that time period. Sprints are used in Scrum methods to support sprint planning, sprint burndown, and other Scrum processes. Sprints are defined via Iteration Paths. Learn more: [About area and iteration paths \(aka sprints\)](#).

Sprint backlog

An interactive list of work items that have been assigned to the same sprint or iteration path for a team. The sprint backlog supports teams that use Scrum methodologies. Learn more: [Sprint planning](#).

Taskboard

An interactive board of work items that support reviewing and updating tasks defined for the sprint backlog. The task board supports teams that use Scrum methodologies. Learn more: [Update and monitor your Taskboard](#).

Teams

A team corresponds to a selected set of project members. With teams, organizations can sub-categorize work to better focus on all the work they're tracking within a project. Each team gets access to a suite of Agile tools. These tools provide teams the ability to work autonomously and collaborate with other teams across the enterprise. Each team can configure and customize each tool to meet their work requirements. Learn more: [About teams and Agile tools](#).

Team Foundation Version Control (TFVC)

A centralized version control system. With TFVC, devs have only one version of each file on their dev machines. Branches are path-based and created on the server. Historical data is maintained only on the server. Branches are path-based and created on the server. Learn more: [Use Team Foundation Version Control](#).

Widgets

Widgets display information and charts on dashboards. Many of them are configurable and display information available from one or more data stores or charts created by the system. Learn more: [Widget catalog](#).

Work items

You plan and track work using work items. Specific types of work items show up on backlogs and boards. Each work item represents an instance of a work item type—such as a user story, feature, bug, or issue—that is assigned a unique identifier, an ID, which is unique across all projects defined in an organization. Learn more: [About work](#)

items.

Work item types (WITs)

A WIT specifies the fields, workflow, and form used to track an item of work. Each WIT is associated with 30+ system fields and several more type-specific fields. You use work items to plan and track the work required to develop your project. For an overview of predefined WITs provided with the default processes, see [Choose a process](#).

2 minutes to read

Troubleshoot connecting to a project in Azure DevOps Services or TFS

9/10/2018 • 4 minutes to read • [Edit Online](#)

Azure DevOps Services | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

Troubleshoot Azure DevOps connectivity

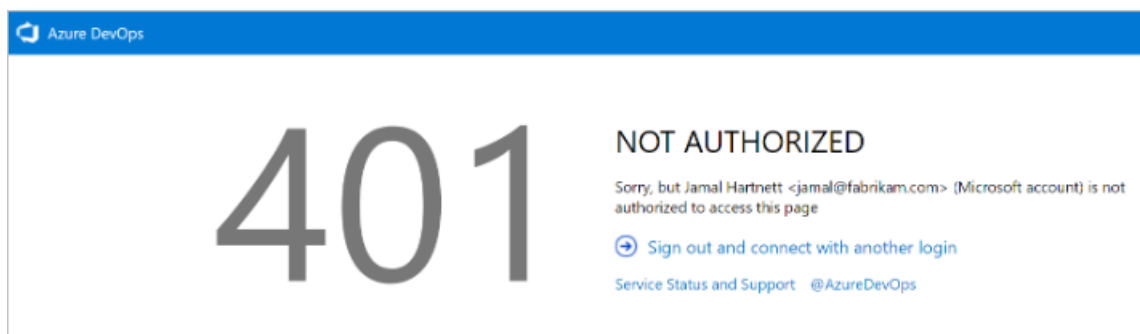
As a first step in resolving connectivity issues with Azure DevOps, complete the following steps:

1. Sign out of your browser. To do this, select the [Visual Studio sign-out](#) link.
2. Delete the cookies in your browser. To delete cookies in most browsers, press 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 to Azure DevOps. If the issue is with the Visual Studio IDE, remove the connection to Azure DevOps, and then re-add it.

Troubleshoot signing in to Azure DevOps

Two types of identities can sign in to Azure DevOps: 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. This occurs when your identity doesn't have permissions to enter an Azure DevOps organization. Common reasons for this include:

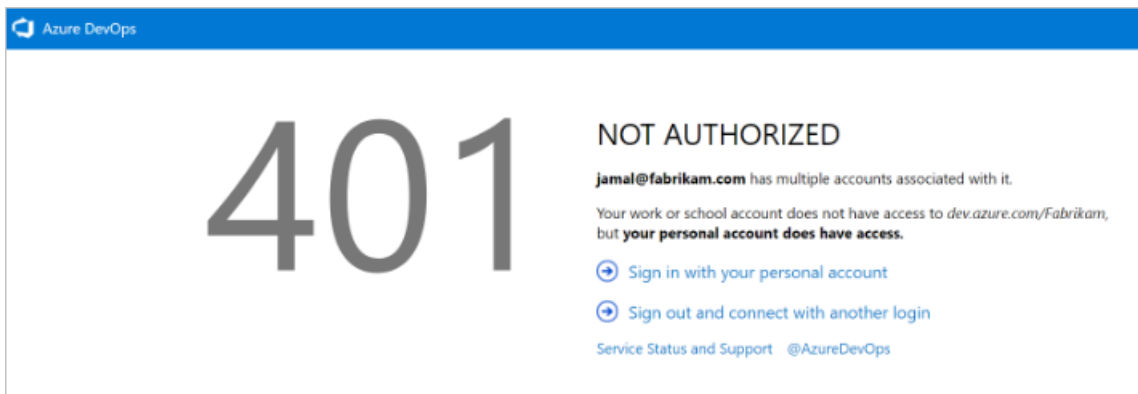
- Your identity isn't a member of the Azure DevOps organization.
- Your identity has an invalid or missing license assignment.

If you think you are a member of the organization but are blocked by this error page, [contact customer 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



This is 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 have signed in with your work or school account, but your personal account is the identity that has access to the Azure DevOps 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 resolve this, select the **Sign in with your personal MSA account** link to sign out of your current work or school Azure AD account, and then sign in by using the personal Microsoft account. After authentication, you should have access to the Azure DevOps organization.

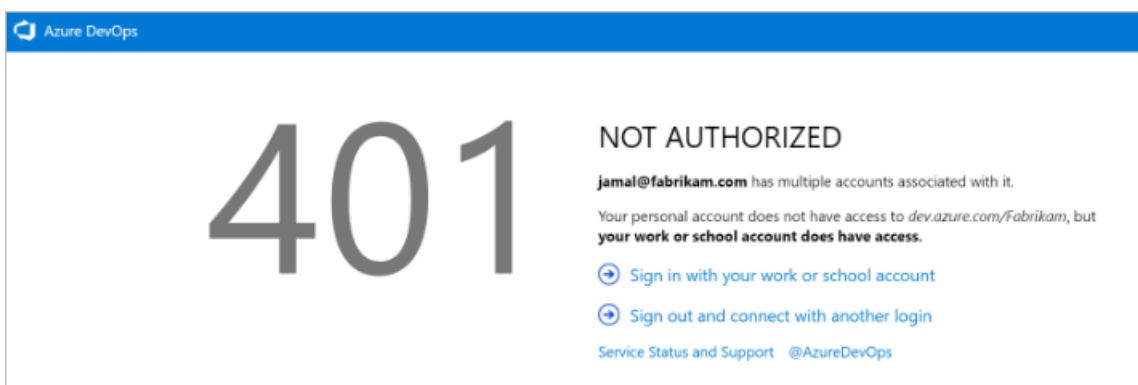
TIP

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 is the opposite version of the 401 error page. In this case, the personal account (Microsoft account identity) doesn't have access to the Azure DevOps 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



Mitigation

If you enter your credentials correctly but instead of redirecting to your Azure DevOps organization you're redirected back to the original sign-in page, we recommend clearing all cookies and then reattempting to sign in. If that doesn't fix the issue, contact customer support.

Troubleshoot TFS connectivity

Here's a list of the most frequently encountered 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 TFS hasn't been taken offline. Talk with your TFS administrator.
4. Check whether your project has been moved to another project collection in TFS. 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 Team Foundation Server](#).

Switch organizations in Azure DevOps

When you use two or more organizations that are linked to Azure AD, such as organizations that are created in the Azure portal, 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 Team Foundation Server** dialog box:

TF31003: Either you have not entered the necessary credentials, or your user account does not have permission to connect to the Team Foundation Server

TF31002: Unable to connect to this Team Foundation 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 Online](#).

Connect to TFS with Secure Sockets Layer

If you connect to a TFS instance that has Secure Sockets Layer (SSL) configured, you must install a certificate and clear the client cache. For details, see [Set up HTTPS with Secure Sockets Layer \(SSL\) for TFS - Configuring client computers](#).

Clear the cache on client computers

When the on-premises TFS configuration changes, such as when you move or split a project collection, you might need to clear the cache.

1. Sign in to your client computer for TFS 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 that's running on the client computer:

- **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 to this Team Foundation Server {0}. Team Foundation Server URL: {1}

9/10/2018 • 5 minutes to read • [Edit Online](#)

Azure DevOps Services | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

You might receive this error when you try to connect to Azure DevOps Services or an on-premises Team Foundation 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 Assign licenses to users for details.
Your Azure DevOps Services organization is connected to the Azure Active Directory.	<p>When your Azure DevOps Services organization is connected to a directory that is associated with an Office 365 or Microsoft Azure subscription, only members in the directory can access the account.</p> <p>Check with your directory administrator to have them create an organizational account for you or add your account to the directory as external member.</p>
You can't switch between different organizational accounts.	<p>If you work with several Azure DevOps Services organizations that connect to different directories, such as accounts that are created from the Microsoft Azure Preview 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.</p> <p>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.</p> <p>To resolve this problem, apply the most recent Visual Studio update.</p> <p>To learn more, see KB Article ID 2958966, You can't switch between different organizational accounts in Visual Studio Online.</p>
You want to sign in to Azure DevOps Services from Visual Studio using different credentials.	See Connect to projects, Sign in with different credentials .

You receive this error when you try to connect to an on-premises TFS 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 outlined below.

PROBLEM	RESOLUTION
Your password has expired.	Verify that you typed your user account and password correctly, and that your password hasn't expired.
You've entered an incorrect server URL.	Verify that you have typed the server URL correctly including the server name, port number, and protocol (http/https). See Connect to projects to learn more.
The TFS configuration has changed.	If the configuration for the on-premises TFS has changed, you must create a new connection. You might also need to clear the client cache .
You work remotely and need to connect to a TFS Proxy server to check in files to Team Foundation version control.	You need to configure Visual Studio to connect to TFS Proxy .
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 Requirements and compatibility for details.
Your firewall is blocking TFS services.	See Allow a program to communicate through Windows Firewall .
Visual Studio stops responding when you run a query in Visual Studio.	Your computer might be configured to bypass the proxy server. You should verify the configuration of the BypassProxyOnLocal setting on your computer. For more information, see BypassProxyOnLocal Configuration .

Several users receive this error when they try to connect to an on-premises TFS

If the problem occurs on more than one computer, you should contact your TFS administrator to confirm whether the server is operational and available on the network.

As an administrator, you should 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. Therefore, you should try the resolutions in the order in which they appear so that you increase 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 Change the service account or password for Team Foundation Server .
The application-tier server for Team Foundation is unavailable.	You should verify whether each required service is running. If a required service is not running, you must restart it. If necessary, set it to start automatically. For more information, see Stop and start services, application pools, and websites .
The network is unavailable.	You should verify whether your network is operational.
A website identity for Team Foundation is configured incorrectly.	You should 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.	You should 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.	You should 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 Change URL 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 Review Server Status and Settings and Verify or Correct Port Assignments .
Trust relationships between domains are not configured correctly.	If a group of users cannot 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 can occur because the GUIDs for the TFS 2012 collection are the same as that for TFS 2008. This confuses the local client cache 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 this, you need to run the TFSSConfig ChangeServerID command. See TFSSConfig ChangeServerID command.</p>

If the previous resolutions do not solve the problem, go to the [MSDN Forums - Visual Studio Team System — Team Foundation Server - Administration](#).

Troubleshoot network connections and whitelist addresses

9/10/2018 • 2 minutes to read • [Edit Online](#)

Azure DevOps

If you are having network connection issues to Azure DevOps, using NuGet, or connecting from Visual Studio 2015 and later versions, it may be because your security appliances are blocking connections now that Visual Studio uses TLS 1.2.

To fix this issue, update the security appliances in order to support TLS 1.2 for the following connections:

List of URLs for sign-in and licensing connections:

- `https://management.core.windows.net`
- `https://login.microsoftonline.com`
- `https://login.live.com`
- `https://go.microsoft.com`
- `https://graph.windows.net`
- `https://app.vsspsxt.visualstudio.com`

A more general list of URLs for signing in to Azure DevOps and Azure:

- `*.windows.net`
- `*.microsoftonline.com`
- `*.visualstudio.com`
- `*.microsoft.com`
- `*.live.com`
- `cdn.vsassets.io` -- hosts our CDN content
- `*.gallerycdn.vsassets.io` -- hosts Azure DevOps extensions
- `static2.sharepointonline.com` -- hosts some resources that we use in "office fabric" UI kit (fonts, etc).

NuGet connections:

- `*.azurewebsites.net`
- `*.nuget.org`

NOTE

Privately owned NuGet server URLs may not be included in the list above. You can check the NuGet servers you are using by opening up `%APPData%\Nuget\NuGet.Config`.

Get Azure DevOps Services and TFS product support and provide feedback

9/10/2018 • 3 minutes to read • [Edit Online](#)

Azure DevOps Services | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

We're always working to improve Azure DevOps Services, and we want you to be part of the process! Share your feedback and ideas with us, or join our communities.

Do you need to do any of the following?:

- **Get advice** Visit StackOverflow for [Azure DevOps Services](#) or [TFS](#).
- **Report a bug** Submit it through our Developer Community for [Azure DevOps Services](#) or [TFS](#).
- **Suggest a feature or a fix** Submit your idea or issue through [UserVoice for Azure DevOps Services](#).
- **Find out what's new in Azure DevOps Services** Check out the [current Azure DevOps Services Release Notes](#). These notes are updated every three weeks.

Azure DevOps Services and TFS Product Support

The primary support venues for Azure DevOps Services and TFS are as follows:

- [Azure DevOps Support](#)
- [TFS Support](#)

For technical support:

- [Basic support](#)
- [Premium support](#)

For billing support:

- [Azure DevOps](#) (from the Azure Support Portal)

Documentation feedback

All docs on docs.microsoft.com 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.

You can add more detailed feedback by clicking the "Tell us more" link after selecting **Yes** or **No**. Check an appropriate box and add what we can do to improve the content for you! Although we cannot reply back, we collect and review these regularly, and use your sentiments in our content planning.

Tips for effective feedback

If you just want to vent about the product or the docs, 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 am using Azure DevOps Services. I am 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 big organization that works on Java apps. I tried to use Maven with your build system in TFS 2017 Update 1 (15.112.26307.0), but I couldn't get the configuration shown in the docs to work."

The more details, the better!

What platform/version am I using?

You can tell what platform you use from the URL you use to connect to Azure DevOps Services or TFS.

Azure DevOps Services

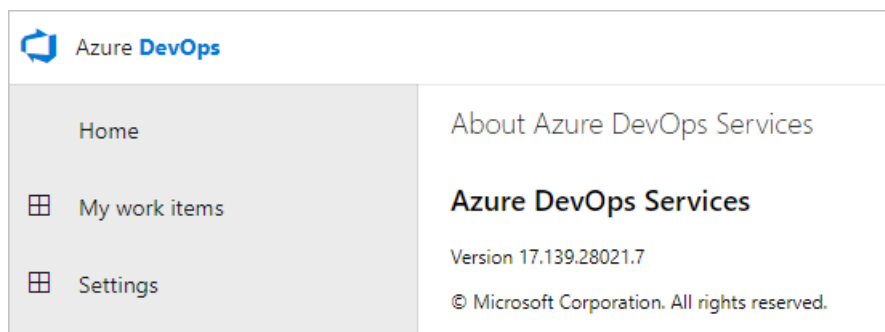
An Azure DevOps Services URL consists of an organization name and dev.azure.com, for example:

```
https://dev.azure.com/{yourorganization} .
```

To learn the version number, enter the following address in a web browser:

```
https://dev.azure.com/{yourorganization}/_home/About
```

A page similar to the one shown will open showing the version number.



TFS

A TFS URL consists of a server name and collection, for example:

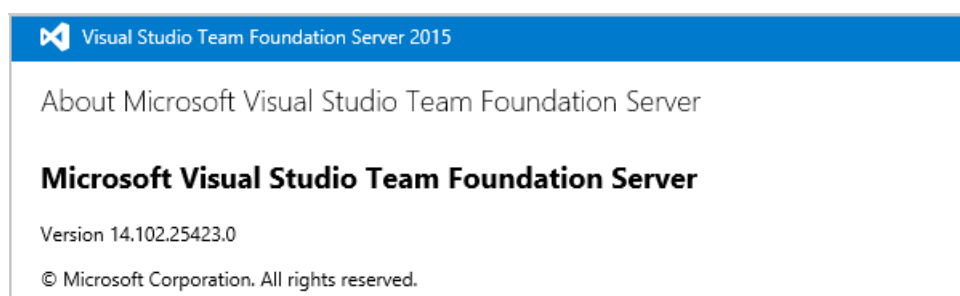
```
https://ServerName:8080/tfs/CollectionName
```

 .

To learn the version number, enter the following address in a web browser:

```
https://ServerName:8080/tfs/_home/About
```

A page similar to the one shown will open showing the version number.



TFS RELEASE	UPDATE	VERSION NUMBER
TFS 2018	RTW	16.122.27102.1
	RC2	16.122.26918.3
	RC1	16.121.26818.0
TFS 2017	Update 3	15.117.27024.0
	Update 3 RC	15.117.26912.0
	Update 2	15.117.26714.0
	Update 1	15.112.26307.0
	RTW	15.105.25910.0
	RC1	15.103.25603.0
	Update 3	14.102.25423.0
	Update 2.1	14.95.25229.0
TFS 2015	Update 2	14.95.25122.0
	Update 2 RC 2	14.95.25029.0
	Update 2 RC 1	14.95.25005.0
	Update 1	14.0.24712.0
	Update 1 RC 2	14.0.24626.0
	Update 1 RC 1	14.0.24606.0
	RTM	14.0.23128.0
	RC2	14.0.23102.0
	RC	14.0.22824.0
	CTP	14.0.22604.0
	Update 5	12.0.40629.0
	Update 4	12.0.31101.0
	Update 4 RC	12.0.31010.0
TFS 2013	Update 3	12.0.30723.0

TFS RELEASE	UPDATE	VERSION NUMBER
	Update 3 RC	12.0.30626.0
	Update 2	12.0.30324.0
	RTM	12.0.21005.1
	RC	12.0.20827.3
TFS 2012	Update 4	11.0.61030.0
	Update 3	11.0.60610.1
	Update 2	11.0.60315.1
	CU 1	11.0.60123.100
	Update 1	11.0.51106.1
	RTM	11.0.50727.1
TFS 2010	CU 2	10.0.40219.371
	SP1	10.0.40219.1
	RTM	10.0.30319.1
TFS 2008	SP1	9.0.30729.1
	RTM	9.0.21022.8
TFS 2005	SP1	8.0.50727.762
	RTM	8.0.50727.147

Related articles

- [Report a problem with Visual Studio](#)

Default permissions and access

9/10/2018 • 8 minutes to read • [Edit Online](#)

Azure DevOps Services | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

To connect and use the functions and features that Azure DevOps Services and Team Foundation Server (TFS) provide, users must be added to a group with the appropriate permissions. The most common built-in groups include Readers, Contributors, and Project Administrators. These groups are assigned the default permissions as listed below.

In addition to permissions, access to specific features are controlled by the access level assigned to a user. Contributors and administrators should be added to Basic (paid) access. Stakeholder access is available to support free access to a limited set of features by an unlimited set of stakeholders.

For a complete reference of all built-in groups and permissions, see [Permissions and groups](#). For information about access levels and supporting stakeholder access, see [About access levels](#).

Boards/Work

You can connect to work items from the web portal, **Boards** or **Work**, and using Eclipse, Visual Studio, Excel, Project, and other clients. For an overview of work tracking features and functions, see [About Agile tools](#).

Stakeholders have limited access to select work tracking functions as described in [Work as a stakeholder](#).

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](#), [queries and query folders](#), and [delivery plans](#).

The team administrator role supports configuration of team settings. To be added as a team administrator, see [Add a team administrators](#).

NOTE

There are no UI permissions associated with [managing tags](#). Instead, you can manage them using the [TFSSecurity command line tool](#).

Users granted Stakeholder access are granted different access to features depending on whether it is a private or a public project. For private projects, Stakeholders have limited access to select work tracking functions, whereas for public projects, Stakeholders enjoy full access to work tracking features. To learn more, see [About access levels](#), [Stakeholder access](#).

TASK	STAKEHOLDERS	READERS	CONTRIBUTORS	TEAM ADMINS	ORGANIZATION OWNER/PROJECT ADMINS
View work items, including bugs, requirements, and tasks	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Create and edit work items, follow a work item	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Change work item type	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Move or delete work items ¹			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Search and query work items, save work item queries	<input type="checkbox"/>	Can't save queries	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
View backlogs, boards, and plans	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Provide feedback	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Request feedback			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Agile tools (Kanban boards, backlogs, sprint planning, portfolio management) ²	limited interactions	view only	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Configure Agile tools, set team defaults ²				<input type="checkbox"/>	<input type="checkbox"/>
Create new work item tags ³	Can assign existing tags		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
View, add, and configure Delivery Plans ⁴		view only	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Customize project information (area paths, iteration paths, and work tracking processes)	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Powerful semantic work tracking search	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Notes:

1. Public project Stakeholders have full access.
2. Public project Stakeholders have full access to all features.
3. Public project Stakeholders can create new tags.
4. Public project Stakeholders can configure Delivery plans.

TASK	STAKEHOLDERS	READERS	CONTRIBUTORS	TEAM ADMINS	PROJECT ADMINS
View work items, including bugs, requirements, and tasks	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Create and edit work items, follow a work item	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Change work item type	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Move or delete work items			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Search and query work items, save work item queries	<input type="checkbox"/>	Can't save queries	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
View backlogs, boards, and plans	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Provide feedback	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Request feedback			<input type="text"/>	<input type="text"/>	<input type="text"/>
Agile tools (Kanban boards, backlogs, sprint planning, portfolio management)	limited interactions	view only	<input type="text"/>	<input type="text"/>	<input type="text"/>
Configure Agile tools, set team defaults				<input type="text"/>	<input type="text"/>
Create new work item tags	Can assign existing tags		<input type="text"/>	<input type="text"/>	<input type="text"/>
View, add, and configure Delivery Plans		view only	<input type="text"/>	<input type="text"/>	<input type="text"/>
Customize project information (area paths, iteration paths, and work tracking processes)	<input type="text"/>		<input type="text"/>		<input type="text"/>
Powerful semantic work tracking search	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Repos/Code

You can connect to your code from the web portal, **Repos** or **Code**, or using Xcode, Eclipse, IntelliJ, Android Studio, Visual Studio, or Visual Studio Code. For an overview of code features and functions, see [Git](#) and [Use Team Foundation Version Control \(TFVC\)](#). Stakeholders for private projects have no access to **Repos** or **Code** features.

From **Project Settings**, you can [set permissions on a repository](#). From the **Code>Branches** page, you can [set permissions for a specific branch and set branch policies](#).

Git

TASK	READERS	CONTRIBUTORS	BUILD ADMINS	ACCOUNT OWNER/ PROJECT ADMINS
Clone, fetch, pull, and explore the contents of a repository	✓	✓	✓	✓
Unlimited private Git repositories		✓	✓	✓
Create branches and tags, manage notes		✓	✓	✓
Create, delete, and rename repositories				✓
Manage permissions, manage branches and branch policies				✓
Powerful semantic code search		✓	✓	✓

TFVC

TASK	READERS	CONTRIBUTORS	BUILD ADMINS	ACCOUNT OWNER/ PROJECT ADMINS
Contribute to a centralized version control, including Code Review (Check in, label, lock, merge, pend a change)	Read only	✓	✓	✓
Check in, revise, undo, unlock other users' changes				✓
Manage branches, manage permissions				✓
Powerful semantic code search		✓	✓	✓

Pipelines/Build and Release

You can define and manage your builds and releases from the web portal, **Pipelines** or **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 build pipelines, release pipelines, task groups, or variable groups. See [Set build and release permissions](#).

NOTE

When the **Free access to Pipelines for Stakeholders** preview feature is enabled for the organization, Stakeholders get access to all **Build and Release** features. This is indicated by the  preview icon shown in the following table. Without this feature enabled, stakeholders can only view and approve releases. To learn more, see [Provide Stakeholders access to edit build and release pipelines](#).

TASK	STAKEHOLDERS	READERS	CONTRIBUTORS	BUILD ADMINS	ORGANIZATION OWNER/ PROJECT ADMINS	RELEASE ADMINS
View build and release pipelines	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Define builds with continuous integration	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Define releases, manage deployments, manage releases with Release Management	<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>
Approve releases	<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>

Package Management (5 users free)	<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>
Queue builds, edit build quality	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Manage build queues and build qualities	<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>	
Manage build retention policies, delete and destroy builds	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Administer build permissions	<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>	
Manage release permissions	<input type="checkbox"/>				<input type="checkbox"/>	<input type="checkbox"/>
Create and edit task groups	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Manage task group permissions	<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Can view library items such as variable groups	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Use and manage library items such as variable groups	<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

TASK	STAKEHOLDERS	READERS	CONTRIBUTORS	BUILD ADMINS	PROJECT ADMINS	RELEASE ADMINS
View build and release pipelines	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Define builds with continuous integration			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Define releases, manage deployments, manage releases with Release Management			<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>
Approve releases	<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>
Package Management (5 users free)			<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>
Queue builds, edit build quality			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Manage build queues and build qualities				<input type="checkbox"/>	<input type="checkbox"/>	
Manage build retention policies, delete and destroy builds			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Administer build permissions				<input type="checkbox"/>	<input type="checkbox"/>	
Manage release permissions					<input type="checkbox"/>	<input type="checkbox"/>
Create and edit task groups			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Manage task group permissions				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Can view library items such as variable groups		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Use and manage library items such as variable groups				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Test Plans/Test

You can define and manage manual tests from the web portal, **Test Plans** or **Test**. For an overview of manual test features and functions, see [Testing overview](#).

You set [test permissions at the project level](#) from **Project Settings>Security**.

TASK	STAKEHOLDERS	READERS	CONTRIBUTORS	ACCOUNT OWNER/ PROJECT ADMINS
Exploratory testing, view test runs		✓	✓	✓
Exploratory testing, create and delete test runs			✓	✓
Provide feedback using the Test & Feedback extension	✓	✓	✓	✓
Request feedback using the Test & Feedback extension			✓	✓
Manage test configurations and test environments			✓	✓
Manage test plans and test suites			✓	✓
Test Manager (purchased separately)			✓	✓

Artifacts/Package Management

You can manage feeds from the web portal, **Artifacts** or **Build and release>Packages**. Feeds have three levels of access: Owners, Contributors, and Readers. Owners can add any type of identity—individuals, teams, and groups—to any access level.

To set permissions, see [Secure feeds using permissions](#).

PERMISSION	READER	CONTRIBUTOR	OWNER
List and restore/install packages	✓	✓	✓
Push packages		✓	✓
Unlist/deprecate packages		✓	✓
Delete/unpublish package			✓
Edit feed permissions			✓
Rename and delete feed			✓

Charts, dashboards, and other web portal features

You can define and manage dashboards from the web portal, **Dashboard**. For an overview of dashboard and

chart features, see [Dashboards](#).

You set [dashboard permissions at the team level](#) from the team dashboard page.

Users granted Stakeholder access are granted different access to features depending on whether it is a private or a public project. For private projects, Stakeholders have limited access to select work tracking functions, whereas for public projects, Stakeholders enjoy full access to work tracking features. To learn more, see [About access levels](#), [Stakeholder access](#).

TASK	STAKEHOLDERS	READERS	CONTRIBUTORS	TEAM ADMINS	ORGANIZATION OWNER/ PROJECT ADMINS
View charts and dashboards	✓	✓	✓	✓	✓
Create work item and test tracking charts ¹			✓	✓	✓
View the project page	✓	✓	✓	✓	✓
Edit the project page ¹					✓
Navigate using the Project pages	✓	✓	✓	✓	✓
Add and configure dashboards ¹			With permissions set	✓	✓

**Notes: **

1. Public project Stakeholders have full access to all features.

TASK	STAKEHOLDERS	READERS	CONTRIBUTORS	TEAM ADMINS	PROJECT ADMINS
View charts and dashboards	✓	✓	✓	✓	✓
Create work item and test tracking charts			✓	✓	✓
View the project page	✓	✓	✓	✓	✓
Edit the project page					✓
Navigate using the Project pages	✓	✓	✓	✓	✓
Add and configure dashboards			With permissions set	✓	✓

Analytics

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.

TASK	STAKEHOLDERS	READERS	CONTRIBUTORS	TEAM ADMINS	ACCOUNT OWNER/ PROJECT ADMINS
View Analytics service		✓	✓	✓	✓
View, edit, and delete a shared Analytics view		✓	✓	✓	✓

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 [TFSSecurity command line tool](#).

TASK	STAKEHOLDERS	READERS	CONTRIBUTORS	TEAM ADMINS	ORGANIZATION OWNER/ PROJECT ADMINS
Set personal notifications or alerts	✓		✓	✓	✓
Set team notifications or alerts				✓	✓
Set project-level notifications or alerts					✓
Participate in Team (chat) rooms ^{1<}			✓	✓	✓
READMEs	See Note 2	✓	✓	✓	✓
View Wikis	✓	✓	✓	✓	✓
Provision or create a Wiki					✓
View the project page	✓	✓	✓	✓	✓
Edit the project page					✓
Navigate using the Project pages	✓	✓	✓	✓	✓
Request feedback		✓	✓	✓	✓
Provide feedback	✓	✓	✓	✓	✓
Powerful semantic code search	✓	✓	✓	✓	✓
Powerful semantic work tracking search	✓	✓	✓	✓	✓

Notes

1. Team (chat) rooms have been deprecated for Azure DevOps Services and TFS 2018 and later versions.
2. Can view project READMEs, but not READMEs defined for a repository.

Related notes

- [Add users to a project or team](#)
- [Permissions and groups reference](#)
- [About access levels](#)
- [Web portal navigation](#)

About access levels

9/10/2018 • 12 minutes to read • [Edit Online](#)

Azure DevOps Services | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

Access levels enable administrators the ability to provide their user base access to the features they need and only pay for those features. To connect and use the functions and features that TFS provides, users must be added to a group with the appropriate permissions. To use select web portal features, they must also belong to the access level that enables access to that feature.

Make sure to set each user's access level based on what you've purchased for that user. Basic access includes all Stakeholder features. Advanced and Visual Enterprise access levels include all Basic features.

To add user accounts or groups to specific access levels, see [Manage users and access](#).

To add user accounts or groups to specific access levels, see [Change access levels](#).

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, which is Basic. This provides most users all the features they need. For a simplified overview of the permissions assigned to the most common groups—Readers, Contributors, and Project Administrators—as well as the Stakeholder access group, see [Permissions and access](#).

The systems employ these access levels:

- **Stakeholders:** provides partial access, can be assigned to unlimited users for free
- **Basic:** provides access to most features
- **VS Enterprise** (TFS 2017.1 and later versions): provides access to premium features
- **Advanced** (TFS 2017 and earlier versions): provides access to premium features

Basic access

Assign **Basic** access to all users with a Visual Studio subscriptions and paid Azure DevOps users, including a TFS client access license (CAL). Basic provides access to most features, except for Test and other premium features.

Control panel **Access levels** Legacy extensions

Export audit log

Stakeholder

Basic (default)

Advanced

VS Enterprise

Access levels

Name	Basic
Features	<div>View My Work Items</div> <div>Standard Features</div> <div>Agile boards</div> <div>Basic backlog and sprint planning tools</div> <div>Request and Manage Feedback</div> <div>Chart Viewing</div> <div>Code</div> <div>Build</div> <div>Administer account</div> <div>Advanced home page</div> <div>Advanced backlog and sprint planning tools</div> <div>Web-based Test Execution</div> <div>View Releases and Manage Approvals</div> <div>Author Release Definitions and Manage Releases</div> <div>Advanced portfolio management</div> <div>Team rooms</div> <div>Chart Authoring</div> <div>Analyze test results and manage machine groups</div>

Stakeholder access

Assign **Stakeholder** access to an unlimited number of users for free.

Stakeholder access grants access to features differently depending on whether you're working from a private or a public project. To learn more about public projects, see [What is a public project?](#).

SERVICE, APPLICATION, OR SETTING	PRIVATE PROJECT	PUBLIC PROJECT
Dashboards	Partial access	Full access
Wiki	Full access	Full access
Boards (Work)	Partial access	Full access
Repos (Code)	No access	Full access
Pipelines (Build and Release)	Full access	Full access
Test Plans (Test)	No access	No access
Notifications	Full access	Full access
Semantic search	Full access	Full access
Project settings	Partial access	Partial access
Organization settings	Partial access	Partial access

Assign **Stakeholder** access to those users who need to enter bugs, view backlogs, boards, charts, and dashboards, but who don't have a TFS CAL. Stakeholders can also view releases and manage release approvals. Stakeholder access is free.

Control panel

Access levels

Legacy extensions

Export audit log

Stakeholder

Basic (default)

Advanced

Access levels

Name

Stakeholder

Features

View My Work Items

Standard Features

Agile boards

View Releases and Manage Approvals

Stakeholder access to user features for private projects

The following features are available to Stakeholders from the web portal for private projects.

BOARDS/WORK	ORGANIZATION, DASHBOARDS, WIKI, AND NOTIFICATIONS
<ul style="list-style-type: none"> - View, create, and modify work items ¹ - View, add, and modify items on backlogs ² - View, and modify items on sprint backlogs ² - View, and modify items on the task board ^{2, 3} - View, and modify items (Kanban) ^{2, 3} - Add tasks to the checklist (Kanban) ^{2, 3} - Follow changes made to work items - View the cumulative flow diagram - View, create, and save queries ⁴ - Submit, view, and change feedback responses - Change work item type 	<ul style="list-style-type: none"> - Work across projects - View project welcome pages ⁵ - View team dashboards - Manage personal notifications - Invite users and assign licenses ⁶ - View wiki pages ⁷ <p>Pipelines/Build & Release</p> <ul style="list-style-type: none"> - All features ⁸

Notes:

1. Can assign existing tags to work items, but not create new tags.
2. Cannot change the backlog priority order (all items are added at the end of the backlog), assign items to an iteration using drag and drop, use the mapping pane or forecasting.
3. Cannot move cards on the board to update status, set the values of fields shown on cards, or set or view team capacity.
4. Can save queries under My Queries but cannot save under Shared Queries.
5. Cannot view markdown README files defined for repositories.
6. Can add users and assign licenses when added to the [Project Collection Administrators](#) group. To learn more, see [Manage users and access](#).
7. Have read-only permissions to wiki pages. These permissions can't be changed.
8. When the **Free access to Pipelines Preview** feature is enabled, Stakeholders gain access to all **Pipeline (Build and Release)** features. If it is disabled, Stakeholders have access to [View releases](#) and [Approve releases](#) only.

Stakeholder access to user features for public projects

From the web portal for private projects, Stakeholders have access to the following features in full, similar to those granted to users who were granted Basic access.



BOARDS/WORK	ORGANIZATION, DASHBOARDS, WIKI, AND NOTIFICATIONS
<ul style="list-style-type: none"> - View, create, and modify work items - View, add, and modify items on backlogs - View, and modify items on sprint backlogs - View, and modify items on the task board - View, and modify items (Kanban) - Add tasks to the checklist (Kanban) - Follow changes made to work items - View the cumulative flow diagram - View, create, and save queries - Submit, view, and change feedback responses - Change work item type 	<ul style="list-style-type: none"> - Work across projects - View and edit project welcome pages - View and manage team dashboards - Manage personal notifications - Invite users and assign licenses ¹ - View and edit wiki pages <p>Pipelines/Build & Release</p> <ul style="list-style-type: none"> - All features ²

Notes:

















1. To add users and assign licenses, stakeholders must be added to the [Project Collection Administrators](#) group. To learn more, see [Manage users and access](#).
2. When the **Free access to Pipelines Preview** feature is enabled, Stakeholders gain access to all **Pipeline (Build and Release)** features. If it is disabled, Stakeholders have access to [View releases](#) and [Approve releases](#) only.

Stakeholder access to administrative features

The following administrative features are granted or denied to users with Stakeholder access by default. Additional features are granted to Stakeholders in public projects.

Items with a  checkmark are granted permission by default. Items with an  indicate that permissions aren't granted and can't be granted to Stakeholders. Members of the Project Collection Administrators or Project Administrators group can grant or deny these permissions for Stakeholders.

Project settings

PERMISSION	PRIVATE PROJECT	PUBLIC PROJECT
Bypass rules on work item updates		
Change process of project		
Create work item tag definition		
Delete and restore work items		
Move work items out of a project		
Permanently delete work items		
Suppress notifications for work item updates		
Agile backlog tools management		

The following permissions to manage area and iteration path settings are granted to Stakeholders by default in both private and public projects:

- Create, delete, and edit child nodes
- Edit work items in this node (area path only)
- View work items in this node (area path only)

The following permissions to manage organization settings are granted to Stakeholders in both private and public projects:

- Administer process permissions
- Create, delete, edit processes
- Delete field from account
- Add and manage users

You can change the permissions granted to Stakeholders. See [Grant or restrict access to select features and functions](#).

Stakeholder access to user features

The following features are available to Stakeholders from the web portal.

WORK	DASHBOARDS, WIKI, AND NOTIFICATIONS
<ul style="list-style-type: none"> - View, create, and modify work items ¹ - View, add, and modify items on backlogs ² - View, and modify items on sprint backlogs ² - View, and modify items on the task board ^{2, 3} - View, and modify items (Kanban) ^{2, 3} - Add tasks to the checklist (Kanban) ^{5, 3} - Follow changes made to work items - View the cumulative flow diagram - View, create, and save queries ⁴ - Submit, view, and change feedback responses - Change work item type 	<ul style="list-style-type: none"> - Work across projects - View project welcome pages ⁶ - View team dashboards - Manage personal notifications - View wiki pages ⁷ <p>Build & Release</p> <ul style="list-style-type: none"> - View releases ⁵ - Approve a release

Notes:

1. Stakeholders can assign existing tags to work items, but not create new tags.
2. Stakeholders cannot change the backlog priority order (all items are added at the end of the backlog), assign items to an iteration using drag and drop, use the mapping pane or forecasting.
3. Stakeholders cannot move cards on the board to update status, set the values of fields shown on cards, or set or view team capacity.
4. Stakeholders can save queries under My Queries but cannot save under Shared Queries.
5. Stakeholders can only view and approve releases.
6. Stakeholders cannot view markdown README files defined for repositories.
7. Stakeholders have read-only permissions to wiki pages. These permissions can't be changed.

Stakeholder access to user features

The following features are available to Stakeholders from the web portal of the listed TFS or later version. Those not annotated are available from all versions. To determine your platform or TFS version, see [Platform and version support](#).

WORK	DASHBOARDS AND NOTIFICATIONS
<ul style="list-style-type: none"> - View, create, and modify work items ¹ - View, add, and modify items on backlogs ² - View, and modify items on sprint backlogs ² - View, and modify items on the task board ^{2, 3} - View, and modify items (Kanban) ^{2, 3} - Add tasks to the checklist (Kanban) ^{2, 3} (TFS 2015.1) - Follow changes made to work items (TFS 2017) - View the cumulative flow diagram - View, create, and save queries ⁴ - Submit, view, and change feedback responses 	<ul style="list-style-type: none"> - Work across projects (TFS 2017) - View project welcome pages ⁶ (TFS 2017) - View team dashboards (TFS 2015) - Manage personal notifications (TFS 2017) - Set personal alerts for changes to work items (TFS 2013, 2015) <p>Build & Release</p> <ul style="list-style-type: none"> - View releases ⁵ (TFS 2015.2) - Approve a release ⁵ (TFS 2015.2)

Notes:

1. Stakeholders can assign existing tags to work items, but not create new tags.
2. Stakeholders cannot change the backlog priority order (all items are added at the end of the backlog), assign items to an iteration using drag and drop, use the mapping pane or forecasting.
3. Stakeholders cannot move cards on the board to update status, set the values of fields shown on cards, or set or view team capacity.
4. Stakeholders can save queries under My Queries but cannot save under Shared Queries.
5. Stakeholders can only view and approve releases.
6. Stakeholders cannot view markdown README files defined for repositories.

Features stakeholders can't access

If you need access to the following features—which support the daily work of product owners, team leads, developers, testers, and project administrators—you need to have **Basic** access.

NOTE

Stakeholders that choose a feature that's not available to them receive an error message indicating that they don't have permissions to complete the task.

For Private projects:

- Change the priority of an item within a backlog
- Delete work items or move work items to another project
- Create shared queries, view charts, and modify the home page
- View Delivery Plans (a Marketplace extension)
- Access the full set of features under **Pipelines (Build and Release)**, **Repos (Code)** or **Test Plans (Test)**.

For Public projects:

- View Delivery Plans (a Marketplace extension)
- Access the full set of features under **Repos (Code)** or **Test Plans (Test)**.
- Change the priority of an item within a backlog
- Delete work items or move work items to another project
- Create shared queries, view charts, and modify the home page
- View Delivery Plans (a Marketplace extension)
- Access the full set of features provided under **Code**, **Build and Release**, and **Test**
- Change the priority of an item within a backlog
- Delete work items or move work items to another project
- Create shared queries, view charts, and modify the home page
- View Delivery Plans (a Marketplace extension)
- Access the full set of features provided under **Code**, **Build and Release**, and **Test**
- Participate in team rooms, which capture interactive, detailed conversations about the project.

VS Enterprise

For TFS 2017.2 and later versions, assign **VS Enterprise** to those users for whom you've purchased Visual Studio Enterprise. These include a TFS CAL plus the rights to access VS Enterprise features. (For users with MSDN Platforms subscriptions or Test Professional, assign the Basic access level and the Test Manager extension.) To learn more, see [Assign paid extension access to users](#). For example, for users with Visual Studio Test Professional or Visual Studio Enterprise, assign them [access to the Test Manager extension](#).

Control panel
Access levels
Legacy extensions

Export audit log
Stakeholder
Basic (default)
Advanced
VS Enterprise

Access levels

Name	VS Enterprise
Features	View My Work Items Standard Features Agile boards Basic backlog and sprint planning tools Request and Manage Feedback Web-based Test Execution View Releases and Manage Approvals Author Release Definitions and Manage Releases Web-based Test Case Management Team rooms Chart Viewing Chart Authoring Code Build Administer account Advanced home page Advanced backlog and sprint planning tools Advanced portfolio management Analyze test results and manage machine groups Microsoft published TFS Extensions Test summary access to stakeholder license users

Advanced

For TFS 2017 and earlier versions, you should assign the **Advanced** level to those users for whom you've purchased the full Test feature set. Here are the purchasing options:

- Higher-level Visual Studio subscriptions: Visual Studio Test Professional, Visual Studio Enterprise, or MSDN Platforms subscriptions. These include a TFS CAL plus the rights to access the full set of Test features.
- A paid Azure DevOps user (which includes a TFS CAL) plus the [Test Manager extension](#).

For TFS 2017.2, Assign **Advanced** access to those users for whom you've purchased MSDN Platforms or Visual Studio Test Professional subscriptions. These include a TFS CAL plus the rights to access Test Manager. To learn more, see [Get extensions for TFS, Assign paid extension access to users](#).

TFS 2017.2

Access levels

Name	Advanced
Features	<ul style="list-style-type: none">View My Work ItemsStandard FeaturesAgile boardsBasic backlog and sprint planning toolsRequest and Manage FeedbackWeb-based Test ExecutionView Releases and Manage ApprovalsAuthor Release Definitions and Manage ReleasesWeb-based Test Case ManagementTeam roomsChart ViewingChart AuthoringCodeBuildAdminister accountAdvanced home pageAdvanced backlog and sprint planning toolsAdvanced portfolio managementAnalyze test results and manage machine groupsTest summary access to stakeholder license users

The Advanced access level will be removed from future versions of Team Foundation Server. Please only use this level for active subscribers with either MSDN Platforms or Visual Studio Test Professional subscriptions. Visual Studio Enterprise subscribers should be added to the Visual Studio Enterprise group instead of Advanced. If you have purchased the Test Manager extension, please manage this in the Users tab within the Team Project in which you made the purchase. [Learn more](#)

TFS 2017.1

NOTE

With TFS 2017.1, the Advanced access level was temporarily disabled. Updating to TFS 2017.2 will re-enable it. If you are on TFS 2017.1 and have users with Visual Studio Test Professional or MSDN Platforms subscriptions, you should assign them Basic access level. In addition, you need to open **Users** for the project collections in which they are a member and [assign them the Test Manager extension](#). To learn more, see [Buy access to TFS or TFS Test](#).

TFS 2017, TFS 2015, TFS 2013

Control panel	Access levels	Legacy Extensions	Agent pools
Export audit log	Access levels		
Stakeholder	Name	Advanced	
Basic (default)	Features	<ul style="list-style-type: none">View My Work ItemsStandard FeaturesAgile boardsBasic backlog and sprint planning toolsRequest and Manage FeedbackWeb-based Test ExecutionView Releases and Manage ApprovalsCreate and Manage ReleasesAuthor Release DefinitionsWeb-based Test Case ManagementTeam roomsChart ViewingChart AuthoringCodeBuildAdminister accountAdvanced home pageAdvanced backlog and sprint planning toolsAdvanced portfolio managementAnalyze test results and manage machine groups	
Advanced			

NOTE

The **Advanced** access level is deprecated for TFS 2017 and later versions of TFS. Use the **VS Enterprise** access level only for active Visual Studio Enterprise subscribers. For MSDN Platforms and Visual Studio Test Professional with MSDN subscribers needing access to **Test**, assign them to the **Advanced** access level and the Test Manager extension.

Test Plans/Test features and Marketplace extensions

Full access to **Test Plans/Test** features requires **VS Enterprise** access. Visual Studio Test Professional plus the test features in the web portal are managed through Azure DevOps, Azure billing services, and purchase of Test Manager extensions from the Marketplace.

To learn more, see [Start free trials for paid Azure DevOps Services features and extensions](#).

Test features and Marketplace extensions

Full access to **Test Plans/Test** features requires **Advanced** (TFS 2015 or earlier versions) or **VS Enterprise** (TFS 2017 or later version) access. Visual Studio Test Professional plus the test features in the TFS web portal are managed through Azure DevOps, Azure billing services, and purchase of Test Manager extensions from the Marketplace.

To learn how to grant access to an extension, see [Get extensions for TFS](#).

What features can users access who are added to two different groups?

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**.

TFS Service account access

[TFS service accounts](#) are added to the default access level. If you make Stakeholder the default access level, you must set the TFS service accounts to Basic or Advanced/VS Enterprise access.

Service accounts don't require a TFS CAL or other purchase.

Related articles

- [Manage users and access](#)
- [Export a list of users and their access levels](#)
- [Default permissions and access](#)
- [Change access levels](#)
- [Export a list of users and their access levels](#)
- [Default permissions and access](#)

Keyboard shortcuts for Azure DevOps Services, TFS, and Team Explorer

9/10/2018 • 10 minutes to read • [Edit Online](#)















Azure DevOps Services | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

You can use the keyboard shortcuts listed in this topic when you work within Azure DevOps Services, the web portal for Team Foundation Server (TFS), or Team Explorer. In addition to these shortcuts, you can [assign your own shortcuts in Visual Studio](#) from the **Tools/Options/Environment/Keyboard** page.

For specific guidance on navigating within the web portal, see [Web portal navigation](#).

Web portal

You can use these keyboard shortcuts when working in the web portal for Azure DevOps or TFS.

Navigate		Navigate within lists	
Ctrl+Alt,a	Move focus to  admin link	Tab	Move focus
Ctrl+Alt,h	Move focus to  help link	→ 	Move focus left/right
Ctrl+Alt,s	Move focus to search box	 	Move focus up/down
Ctrl+Alt, 	Move focus to next section	Ctrl+Home	Move focus to top of list
Ctrl+Alt, 	Move focus to previous section	Ctrl+End	Move focus to bottom of list
		Ctrl+  	Move item up/down within list
		Shift  	Highlight consecutive items
		Menu	Open context menu
		Esc	Dismiss context menu
		→ 	Move focus left/right
		 	Move focus up/down
		Enter	Choose selected menu item

Web portal, global shortcuts

Type **?** to access the Global and page-specific shortcuts.

NOTE

The following shortcuts are available for TFS 2015.2 and later versions. Type **?** to access the Global and page-specific shortcuts.

You can use the following keyboard shortcuts from the web portal.

	<p>? Show shortcuts</p> <p>p Projects and teams</p> <p>g,h Go to Dashboards</p> <p>g,w Go to Boards or Work</p> <p>g,c Go to Repos or Code</p> <p>g,b Go to Pipelines or Build and release</p> <p>g,t Go to Test Plans or Test</p> <p>g,s Go to [Project Settings] (/azure/devops/organizations/settings/about settings)</p> <p>f,n Focus next section</p> <p>f,p Focus previous section</p> <p>s Move focus to search</p>
--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Page-specific shortcuts only work when in a specific page. For example, type **g c** to open the **Code** page, and then type **c p** to create a pull request. These navigation shortcuts work as long as the focus is not on an input control.

Code

You can use the following keyboard shortcuts when working from a page under **Repos** or **Code**. To view the valid shortcuts, enter **Shift+?** to access Global and service-specific shortcuts..

NOTE

The following shortcuts are available from the web portal for Azure DevOps Services and TFS 2015.2 and later versions.

	<p>Code</p> <p>r Select repository</p> <p>e Open explorer</p> <p>h Open history</p> <p>b Open branches (Git)</p> <p>q Open pull requests (Git)</p> <p>c,p Create pull request (Git)</p> <p>c Open changesets (TFVC)</p> <p>v Open shelve sets (TFVC)</p> <p>File Explorer</p> <p>1 Open contents</p> <p>2 Open history</p> <p>t Move focus to directory path</p> <p>w Select branch (Git)</p> <p>y Switch to commit (Git)</p> <p>c,b Create branch (Git)</p>
--	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Work Items

You can use the following keyboard shortcuts when working from the **Repos>Work Items** or **Work>Work Items** page.

NOTE

The following shortcuts are available from the web portal for Azure DevOps Services.

<input type="text"/>	Work <ul style="list-style-type: none">l Open backlogb Open boardi Open current iterationt Open task boardq Open queriesz Toggle full screen Work items <ul style="list-style-type: none">Ctrl+Shift+f Filter resultsCtrl+c Copy to clipboardDelete Delete
----------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Work item form shortcuts

You can use the following keyboard shortcuts when [interacting with a work item form](#). To view the valid shortcuts, enter **Shift+?** from within the form.

NOTE

Feature availability: The following shortcuts are available from Azure DevOps Services.

<input type="text"/>	Work Item Form <ul style="list-style-type: none">Alt+i Assign work item to meCtrl+Shift+d Go to discussionCtrl+s Save changesShift+Alt+c Copy work item titleCtrl+Shift+, Move to left tab (page)Ctrl+Shift+. Move to right tab (page)
----------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Also, you can use the following keyboard shortcuts when working within the work item form in the web portal, both Azure DevOps Services and TFS.

FORMAT, COPY, PASTE RICH TEXT	SAVE AND CLOSE	
<ul style="list-style-type: none">Ctrl+b Bold textCtrl+i Italicize textCtrl+u Underscore textCtrl+Spacebar Clear formattingCtrl+c Copy textCtrl+v Paste copied text	<ul style="list-style-type: none">Ctrl+s Save changesCtrl+Enter Save and closeEsc Close work item	

Backlogs

You can use the following keyboard shortcuts when working from a **Boards>Backlogs** or **Work>Backlogs** page.

These shortcuts work when you are on a product backlog, portfolio backlog, or sprint backlog page.

<div></div>	<div><div>Backlogs</div><div><div>Ctrl+Home</div><div>Move item to top</div></div><div><div>m,b</div><div>Move item to backlog</div></div><div><div>m,i</div><div>Move item to current iteration</div></div><div><div>m,n</div><div>Move item to next iteration</div></div><div><div>n</div><div>Open new item panel</div></div><div><div>Ins</div><div>Add child</div></div><div><div>Ctrl+Shift+f</div><div>Filter results</div></div><div><div>r</div><div>Show/Hide Parents</div></div></div>
-------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Boards

You can use the following keyboard shortcuts from any Kanban board, that is, when working from **Repos>Boards** or **Work>Board** page.

NOTE

The following shortcuts are available from the web portal for Azure DevOps Services and TFS 2015.2 and later versions.

<div></div>	<div><div>Kanban Board</div><div><div>n</div><div>Add new item</div></div><div><div>c</div><div>Add new child item</div></div><div><div>Home</div><div>Select first item</div></div><div><div>Enter</div><div>Open item</div></div><div><div>Ctrl+Shift+f</div><div>Filter results</div></div><div><div>Ctrl+</div><div></div><div>Move item up</div></div><div><div>Ctrl+</div><div></div><div>Move item down</div></div><div><div>Ctrl+</div><div></div><div>Move item left</div></div><div><div>Ctrl+</div><div></div><div>Move item right</div></div><div><div>Ctrl+Home</div><div>Move item to top of column</div></div><div><div>Ctrl+End</div><div>Move item to bottom of column</div></div><div><div>Ctrl+Shift+</div><div></div><div>Move item to swimlane above</div></div><div><div>Ctrl+Shift+</div><div></div><div>Move item to swimlane below</div></div><div><div>F2</div><div>Rename item</div></div><div><div>e</div><div>Show/hide empty fields</div></div><div><div>o</div><div>Expand all swimlanes</div></div><div><div>u</div><div>Collapse all swimlanes</div></div><div><div>Shift+Pageup</div><div>Select first/next swimlane above</div></div><div><div>Shift+Pagedown</div><div>Select last/next swimlane below</div></div></div>
-------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Queries

You can use the following keyboard shortcuts when [working with queries](#) in the web portal. To view the valid

shortcuts, enter **Shift+?** from the **Work>Queries** page.

NOTE

Feature availability: The following shortcuts are available from Azure DevOps Services or TFS 2015.2 or later versions.

<div></div>	<div>Queries</div> <div><div>c q</div>Add new query</div> <div><div>r or Alt+r</div>Refresh query</div> <div><div>Alt+q</div>Return to query</div> <div><div>j or Alt+n</div>Select next item</div> <div><div>k or Alt+p</div>Select previous item</div> <div><div>Ctrl+Shift+f</div>Filter results</div>
-------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Plans

You can use the following keyboard shortcuts when [interacting with a delivery plan](#). To view the valid shortcuts, enter **Shift+?** when viewing a plan from the **Work>Plans** page.

NOTE

The following shortcuts are available from the web portal for Azure DevOps Services and TFS 2017.2 and later versions. Type **?** to access Global and service-specific shortcuts.

Work	Delivery plan	
<div><div>l</div>Open backlog</div> <div><div>b</div>Open board</div> <div><div>i</div>Open current iteration</div> <div><div>t</div>Open task board</div> <div><div>q</div>Open queries</div> <div><div>z</div>Toggle full screen mode</div>	<div><div>Home</div>Select first item</div> <div><div>Enter</div>Open item</div> <div><div>n</div>New item</div> <div><div>Ctrl+<div></div></div>Move item up</div> <div><div>Ctrl+<div></div></div>Move item down</div> <div><div>Ctrl+<div></div></div>Move item left</div> <div><div>Ctrl+<div></div></div>Move item right</div>	<div><div>Shift+<div></div></div>Pan timeline left</div> <div><div>Shift+<div></div></div>Pan timeline right</div> <div><div>u</div>Collapse all backlogs</div> <div><div>o</div>Expand all backlogs</div> <div><div>Shift+pageup</div>Focus on previous team</div> <div><div>Shift+pagedown</div>Focus on next team</div> <div><div>Ctrl+Shift+f</div>Filter results</div>

Test Plans, Parameters, and Runs

You can use the following keyboard shortcuts when working in **Test Plans** or **Test**.

NOTE

The following shortcuts are available from the web portal for Azure DevOps Services and TFS 2015.2 or later versions.

	Test <ul style="list-style-type: none"> n Open test plans m Open shared parameters r Open runs h Open machines
	Test plan <ul style="list-style-type: none"> 1 Open tests 2 Open charts e Execute tests t,b Mark selected tests as blocked t,f Fail selected tests t,n Mark selected tests as NA t,p Pass selected tests t,r Reset tests to active Ctrl+Shift+f Filter results v,g View grid
	Parameters <ul style="list-style-type: none"> 1 View parameter set grid 2 Open properties c,s Add parameter set c,t Add test case v,t Toggle test cases pane
	Test runs <ul style="list-style-type: none"> 1 Test runs 2 Filter

Wiki

NOTE

Keyboard shortcuts to manage Wiki pages are supported on TFS 2018.2 or later versions. To download TFS 2018.2, see [Team Foundation Server 2018 Update 2 Release Notes](#).

You can use the following keyboard shortcuts when [managing or editing Wiki pages](#). To view the valid shortcuts, enter **Shift+?** from a **Wiki** page.

NOTE

The following shortcuts are available from the web portal for Azure DevOps Services and TFS 2018.2 and later versions.

<input type="text"/>	Wiki (manage) n Add new page e Edit page c Add new sub-page Ctrl+ <input type="text"/> Move page up the order Ctrl+ <input type="text"/> Move page down the order Ctrl+Shift+f Filter page
<input type="text"/>	Wiki edit Ctrl+b Bold text Ctrl+i Italicize text Ctrl+k Insert hyperlink Ctrl+c Copy text Ctrl+v Paste copied text Ctrl+s Save changes Ctrl+Enter Save and Close Esc Close

Team Explorer navigational shortcuts

Use these shortcuts when working in Team Explorer.

Navigate Ctrl+0,a Open web portal Ctrl+0,b Open Build Ctrl+0,c Open Connect Ctrl+0,d Open Documents Ctrl+0,e Open Branches (Git) Ctrl+0,g Open Changes (Git) Ctrl+0,h Open Home Ctrl+0,m Open My Work (TFVC) Ctrl+0,p Open Pending changes (TFVC) Ctrl+0,r Open Reports Ctrl+0,s Open Settings Ctrl+0,w Open Work items Ctrl+0,y Open Synchronization (Git) Ctrl+' Move focus to search box Alt+0 Move focus to top of page Alt+1...9 Move focus to visible section [1 thru 9] Alt+ Move focus to next/previous section	Context menu Open a context menu Esc Dismiss a context menu Move focus left/right Move focus up/down Enter Choose Context menu Work item commands Alt+m,g Open work item Alt+m,i Add a work item Alt+m,q Add a query Shift+Alt,c Copy selected work item Shift+Alt,l Link to new work item Enter Open selected work item
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

You can use query results shortcuts whenever you have a list of work items, such as the query results view or a list of linked work items within a work item form.

QUERY EDITOR	ACTION	QUERY RESULTS	ACTION
<input type="text"/>	Move focus left/right	<input type="text"/>	Scroll left/right
<input type="text"/>	Move focus up/down	PgUp/PgDn	Scroll up/down
Shift+ <input type="text"/>	Highlight consecutive clauses	Shift+ <input type="text"/>	Highlight consecutive rows

QUERY EDITOR	ACTION	QUERY RESULTS	ACTION
Shift+ <input type="text"/>	Move focus left one field at a time	Shift+Alt,n	Move focus to next item
Shift+ <input type="text"/>	Move focus right one field at a time	Shift+Alt,p	Move focus to previous item
End	Move focus to end of current clause	End	Move focus to bottom of list
Enter	Move focus down	Enter	Open selected work item
Tab	Move focus right, one field at a time	Home	Move focus to top of list
Ctrl+c	Copy selected clause	+/-	Expand/collapse current row
Ctrl+s	Save changes (editor)	Ctrl+s	Save changes (results)
Ctrl+v	Paste copied clause	F5	Refresh
Del	Delete contents of current field or clause		

Related articles

- [Keyboard shortcuts for Microsoft Test Manager](#)
- [Customize Visual Studio keyboard shortcuts](#)
- [Default keyboard shortcuts for Visual Studio](#)
- [Accessibility Features of Visual Studio](#)
- [Web portal navigation](#)

Install Team Explorer

Team Explorer is a plug-in to Visual Studio. By installing the free [Visual Studio Community](#), other Visual Studio version, or Visual Studio Team Explorer 2017 you gain access to Team Explorer.

Learn more about [working in Team Explorer](#).

Navigate in Team Explorer

9/10/2018 • 6 minutes to read • [Edit Online](#)

Azure DevOps Services | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

You use Team Explorer to manage work that is assigned to you, your team, or your projects, and to coordinate your efforts with other team members to develop a project. Team Explorer is a plug-in that installs with Visual Studio or Eclipse. Working from different platforms, developers and stakeholders can effectively collaborate using Team Explorer connected to projects hosted on Azure DevOps Services or on-premises Team Foundation Server (TFS).

TIP

You can access 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](#).

The operations available to you depend on which source control option-Team Foundation version control (TFVC) or Git-was selected to manage source code when the project was created.

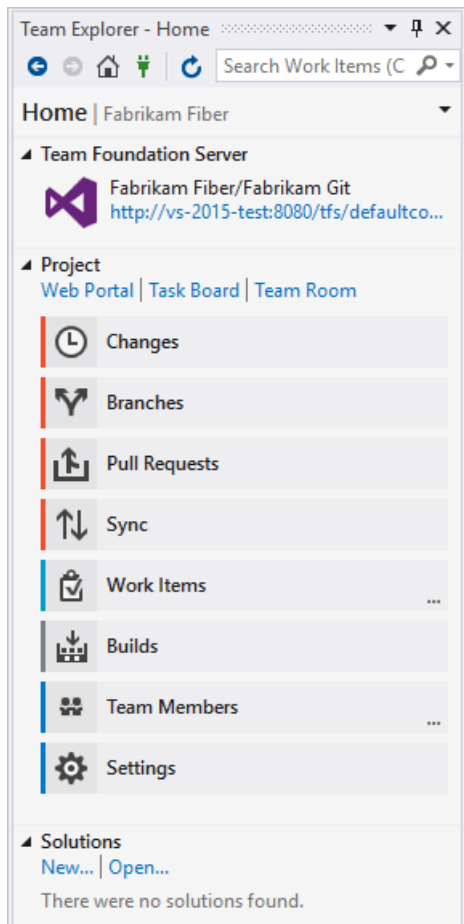
Team Explorer plug-in for Visual Studio

Team Explorer connects Visual Studio to projects. You can manage source code, work items, and builds. Or, create a project.

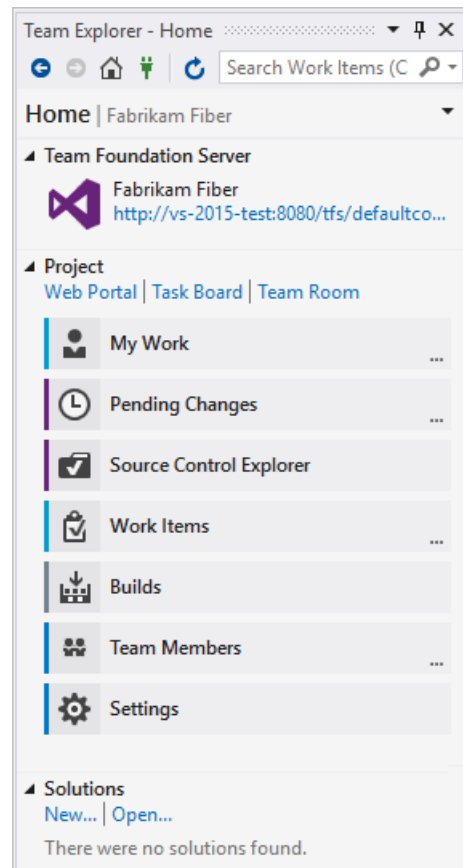
TIP

If you open Visual Studio and the Team Explorer pane doesn't appear, choose the **View/Team Explorer** menu option.

HOME PAGE WITH GIT



HOME PAGE WITH TFVC



NOTE

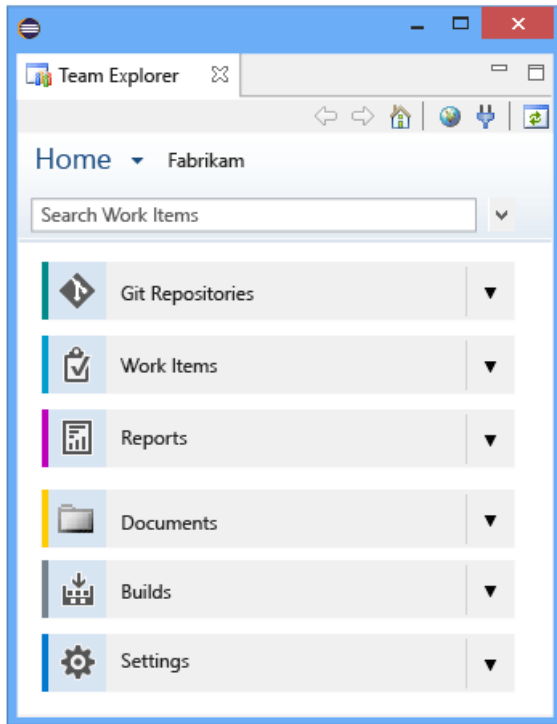
Some pages, such as **Reports** and **Documents**, only appear when an on-premises TFS is configured with the required resources, such as SQL Server Reporting Services and SharePoint.

If you don't need Visual Studio, but do want to connect to Azure DevOps Services or TFS or get one or more Team Foundation add-ins, you can install the free [Visual Studio Community](#).

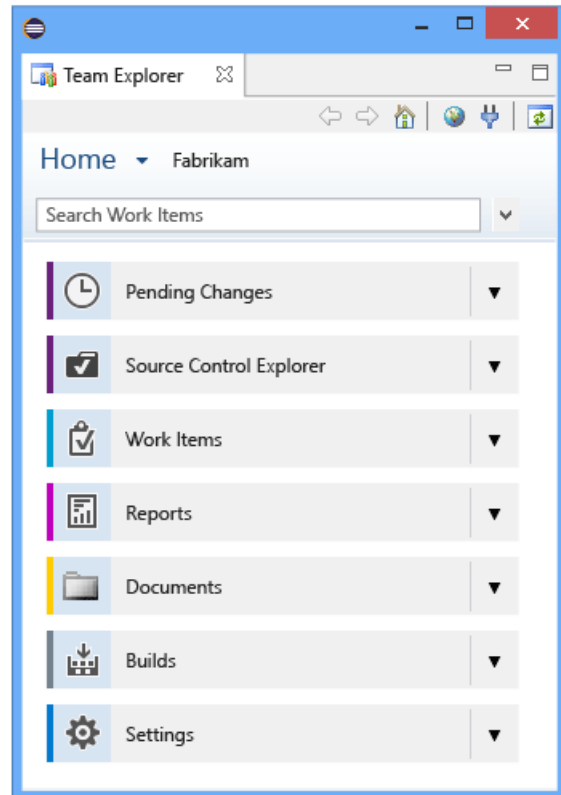
Team Explorer plug-in for Eclipse

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)

**NOTE**

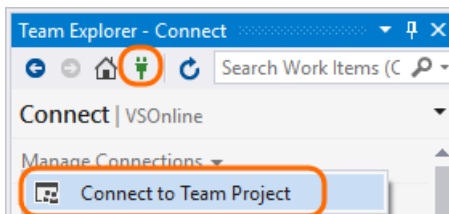
Some pages, such as **Reports** and **Documents**, only appear when TFS is configured with the required resources, such as SQL Server Reporting Services and SharePoint.

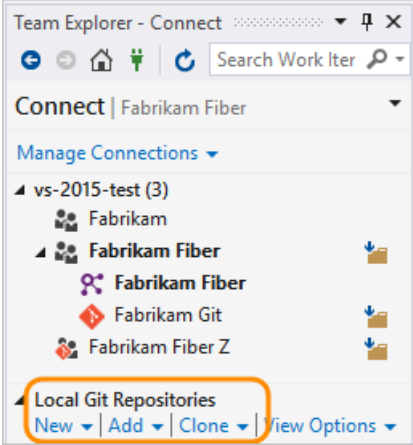
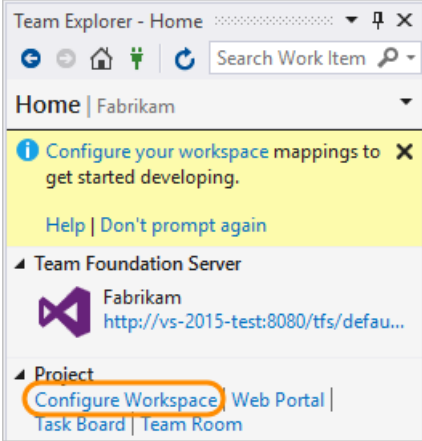
Choose the page to access the task you want

Based on the page you select and the options configured for your project.

Connect page

From the **Connect** page, you can select the projects you want to connect to and quickly switch context between projects.



GIT: LOCAL GIT REPO	TFVC: MAP WORKSPACE
<p>If you connect to a Git repo, you also can create, add, or clone repositories.</p> 	<p>If you connect to a TFVC repo, you'll be prompted to Configure your workspace (TFVC) on first connect.</p> 

IMPORTANT

From the Visual Studio plug-in, you can [Create a project](#). The ability to create projects is not supported from the Eclipse plug-in. You can, however, create projects from the web portal/collection administration context.

Home, Work, and Build pages

HOME	WORK ITEMS	BUILD
<ul style="list-style-type: none"> - Configure workspace - Open Web portal - Open Task Board - Open Team Room 	<ul style="list-style-type: none"> - Add work items - Use the query editor to list and manage queries - Organize query folders and set query permissions - Open query in Excel - Open query in Project - Email query results list using Outlook - Create reports from query in Excel (TFS only) 	<ul style="list-style-type: none"> - Create build pipelines - View and manage builds - Manage the build queue

NOTE

If inline images aren't displaying correctly, see [Resolve images that don't display in Team Explorer](#).

Git and TFVC pages

The Git and TFVC repos support different pages and functions. You'll see one or the other pages depending on the project and repo you connect to. For a comparison of the two repos, see [Choosing the right version control for your project](#).

GIT	TFVC
<ul style="list-style-type: none"> - Changes: Save work with commits - Branches: Create work in branches - Pull Requests: Review code with pull requests - Sync: Update code with fetch and pull 	<ul style="list-style-type: none"> - My Work: Suspend/resume work Code review - Pending Changes: Manage pending changes Find shelvesets Resolve conflicts - Source Control Explorer: Add/view files and folders

Report and Document pages

The **Report** 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.

From the **Document** page, you can [open project portal](#) and [manage documents and document libraries](#). This page appears only if your project has been configured with a SharePoint Products portal.

If your project is missing one or more pages, you may be able to [add functionality to your on premises TFS deployment](#).

Settings page

From the **Settings** page, you can configure administrative features for either a project or project collection. Configuring features in these areas requires you to be a member of an administrator group or have the required permissions.

Most of the links open to a web portal administration page. Not all settings are available from the Team Explorer plug-in for Eclipse.

To learn more about settings, see [About team, project, and organizational-level settings](#).

Clients that connect to Azure DevOps Services and TFS

In addition to connecting through Team Explorer, you can connect to a project from these clients:





- [Web portal](#)
- [Visual Studio Code](#)
- [Visual Studio Community](#)
- [Eclipse: Team Explorer Everywhere](#)
- [Office Excel](#)
- [Office Project](#)
- [PowerPoint Storyboarding](#)
- [Microsoft Test Manager](#)
- [Microsoft Feedback Client](#)

Related articles

- [Troubleshoot connection](#)

Refresh Team Explorer

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** icon in the menu bar (or choose the F5 key).
- To refresh the project you currently have selected, choose  **Home**, and then choose  **Refresh** icon (or choose the F5 key).
- To refresh the set of teams defined for the project that you currently have selected, choose the Connect icon, and then choose  **Refresh** icon (or choose the F5 key).

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 in-line image fails to display in a work item form you view from Visual Studio Team Explorer, but does display in the web portal, your credentials may have expired.

You can resolve it with the following steps.

1. In Visual Studio, click **View>Other Windows>Web Browser** (Or, use the shortcut **Ctrl+Alt+R**).
2. In the web browser, locate your account.
3. Login with your account.
4. Refresh your work item in Team Explorer.

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.

FAQs

9/10/2018 • 2 minutes to read • [Edit Online](#)

Azure DevOps Services | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

How do I get started?

To get started in the cloud or on-premises:

- To get started with Azure DevOps Services begin by [creating a user](#). Step-by-step instructions are provided in [Sign up for Azure DevOps Services](#).
- To get started with an on-premises TFS, download and install the [latest version of TFS](#). See [Install and configure TFS](#) for details.
- If you need to create a project, create one in [Azure DevOps Services](#) or set one up in an [on-premises TFS](#).
- If you don't have access to the project, [get invited to the team](#).
- If it's your first time connecting to a project, see [Connect to a project](#).

To get started with a client tool: Go to one of these pages to download a version of Visual Studio or client tool plug-in that will support connecting to a project:

- [Visual Studio](#)
- [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](#)

To get started with sharing code, work item tracking, builds, or other tasks: See [Software development roles](#).

What compatibility issues exist between client and server versions?

See [Requirements and compatibility](#).

Can stakeholders who don't use Visual Studio participate on our team?

Yes. You can provide access to stakeholders who have no CAL for the following activities:

- **Stakeholder access:** This view allows anyone on your team to check project status and provide feedback. Stakeholders can [track project priorities and provide direction, feature ideas, and business alignment to a team](#).

To grant stakeholders access, add them to the [Stakeholder access group](#).

- **Provide feedback:** To allow your stakeholders to provide feedback, you must [grant them specific permissions](#).

Are there other clients that connect to Azure DevOps Services or TFS? Are there other tools I can use?

Yes. You can connect to a project from one of these clients:

- [Excel](#) (Requires the Team Foundation add-in is installed)
- [Project](#) (Requires the Team Foundation add-in is installed)
- [Project Professional](#)
- [PowerPoint Storyboarding](#) (Requires the Team Foundation add-in is installed)
- [Microsoft Test Manager](#)

- [Test & Feedback extension](#) (previously called the Exploratory Testing extension)
- [Microsoft Feedback Client](#)

NOTE

Native support for integrating TFS with Project Server is deprecated for TFS 2017. However, synchronization support is provided by a third part. See [Synchronize TFS with Project Server](#) for details.
Test Manager is deprecated for TFS 2017.

Also, you can find several open-source clients that have been added to [Marketplace extensions](#). For example, you can install extensions to Visual Studio that support additional features:

- For TFS 2017 and later versions, you can [install the TFS Process Template editor from the Visual Studio Marketplace](#). You can use this version of the Process Editor to modify the old-style work item forms. You can't use it to edit forms associated with the [new web forms](#).
- For TFS 2015 and earlier versions, you can install [TFS Power Tools](#) which provides enhancements, tools, and command-line utilities that support increased productivity.

NOTE

Team Foundation Server Power Tools is deprecated for TFS 2017 and later versions.

Related articles

- [Key concepts](#)
- [Essential services](#)
- [Client-server tools](#)
- [Software development roles](#)

Have more questions? Search for an answer or pose a question in one of the community forums listed in [Provide product and content feedback, Platforms and version support](#).

Web portal navigation

9/10/2018 • 6 minutes to read • [Edit Online](#)

Azure DevOps Services | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

The web portal for Azure DevOps Services and Team Foundation Server (TFS) 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 the navigation selected:

- **New navigation:** **Overview, Boards, Repos, Pipelines, Test Plans, and Artifacts**
- **Previous navigation:** **Dashboards, Code, Work, Build and Release, Test, Wiki, and Analytics views**

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 (TFS) 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.

NOTE

Choose **Previous navigation** when you see a top-level blue bar. Choose **New navigation** if you see a vertical sidebar or if you enabled the **New Navigation** preview feature. The vertical sidebar, along with other navigational features, is enabled when the **New Navigation** preview feature has been enabled for the signed-in user or the organization. To learn how to use the web portal effectively, see [Web portal navigation](#).

For on-premises TFS, choose **Previous Navigation** for guidance.

Here's what you need to know to get up and running using the web portal.

- [New navigation](#)
- [Previous navigation](#)
- **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:** 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.

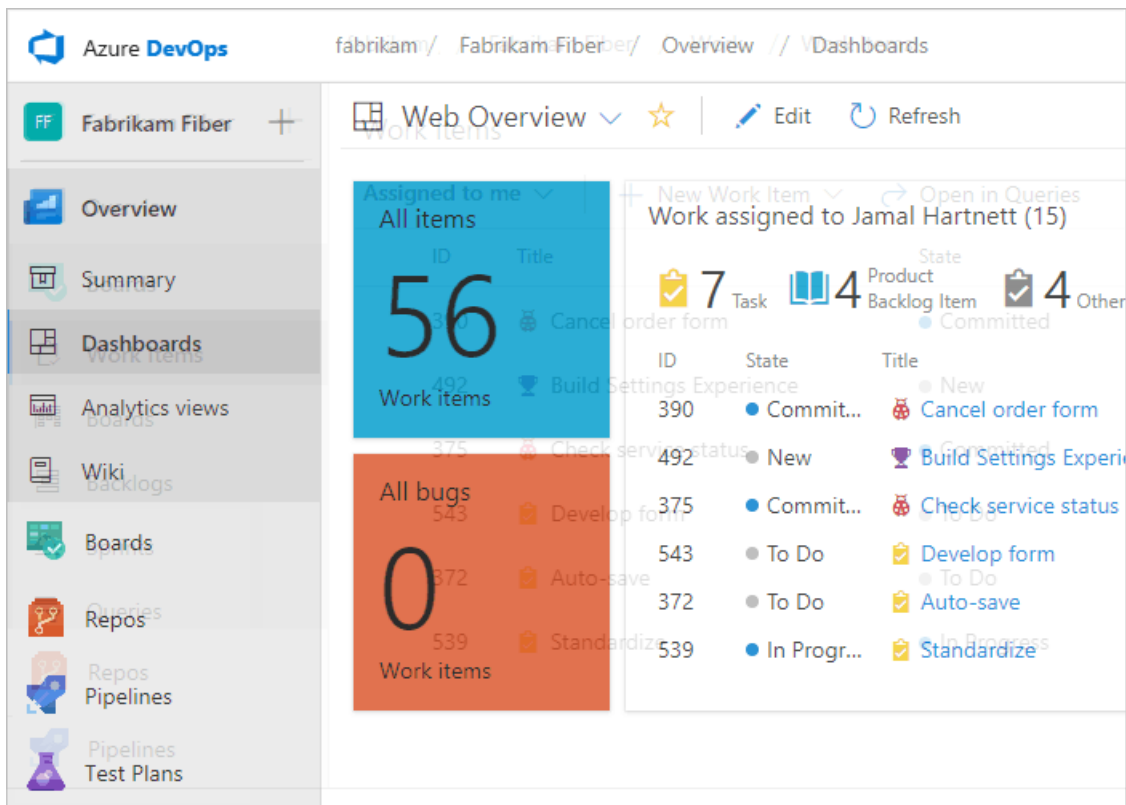
New navigation isn't supported on TFS at this time. Choose **Previous navigation** for guidance.

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](#).

- [New navigation](#)
- [Previous navigation](#)

In **New navigation**, you select services—such as **Boards**, **Repos**, and **Pipelines**—from the sidebar and pages within those services.



New navigation isn't supported on TFS at this time. Choose **Previous navigation** for guidance.

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 the [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

To connect to the web portal, you need your user account added to a project. 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](#).

To connect to the web portal, you need your user account added to a project. This is typically done by a [project administrator](#).

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.

WEB PORTAL	VISUAL STUDIO
<ul style="list-style-type: none">• Product backlog, Portfolio backlogs, Sprint backlogs, Task boards, Capacity planning• Kanban board• Dashboards, Widgets, and Charts• Team rooms• Request feedback• Web-based Test Management• Administration pages to administer accounts, team projects, and teams	<ul style="list-style-type: none">• Task specific interfaces that integrate with Git and TFVC, such as:<ul style="list-style-type: none">◦ 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.

Resources

- [Manage projects](#)
- [Project & Organizational Settings](#)

Manage projects

9/10/2018 • 2 minutes to read • [Edit Online](#)

Azure DevOps Services | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

Structure your projects by adding area paths, iteration paths, and teams.

5-Minute Quickstarts

- [Get started as an administrator](#)
- [Share your project vision](#)
- [Define area paths](#)
- [Define iteration paths or sprints](#)
- [Add a team](#)
- [Add users to a project or team](#)
- [Add administrators or set permissions at the project or collection level](#)

Step-by-Step Tutorials

- [Change individual permissions, grant select access to specific functions](#)
- [Grant or restrict permissions to select tasks](#)
- [Customize a project](#)

Concepts

- [About areas and iterations](#)
- [About teams and Agile tools](#)
- [Resources granted to project members](#)

How-to Guides

- [Create a project](#)
- [Rename a project](#)
- [Change service visibility](#)
- [Connect to projects](#)

Reference

- [Default permissions and access](#)
- [Permission lookup guide \(Security\)](#)

Resources

- [New User Guide](#)
- [Public Projects](#)
- [Security & identity](#)
- [Migrate from TFS to Azure DevOps Services](#)

Security & identity

9/10/2018 • 2 minutes to read • [Edit Online](#)

Azure DevOps Services | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

For anyone to access a project, you must add them to a security group. For a quick look at what permissions are assigned to the default security groups, see [Default permissions and access assignments](#).

5-Minute Quickstarts

- [View permissions](#)
- [Look up the organization owner or a project administrator](#)
- [Add users to a project or team](#)
- [Set Git or TFVC repository permissions](#)
- [Add administrators or set permissions at the project or collection level](#)

Tutorials

- [Set up Active Directory or Azure Active Directory](#)
- [Add AD/Azure AD security groups to built-in security groups](#)
- [Change individual permissions, grant select access to specific functions](#)
- [Grant or restrict permissions to select tasks](#)
- [Remove user accounts](#)

Concepts

- [About permissions and groups](#)
- [About security roles](#)
- [About access levels](#)
- [Azure Active Directory groups \(Azure DevOps\)](#)
- [Active Directory groups \(TFS\)](#)
- [Security glossary](#)

How-to Guides

- [Set Git branch permissions](#)
- [Set build and release permissions](#)
- [Set permissions and access for work tracking](#)
- [Change access levels \(TFS\)](#)

Reference

- [Default permission and access assignments](#)
- [Permissions lookup guide](#)
- [Permissions and groups reference](#)

Resources

- [Account Management \(Azure DevOps\)](#)
- [Server Administration \(TFS\)](#)
- [Billing](#)
- [Authentication guidance for REST APIs](#)
- [Azure DevOps Data Protection Overview](#)
- [Technical Articles](#)

Billing

9/10/2018 • 2 minutes to read • [Edit Online](#)

Azure DevOps Services | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

Billing for Azure DevOps

All Azure DevOps charges appear on your monthly Azure bill. Azure supports payment by credit card as well as invoiced billing through the Enterprise Agreement (EA), Cloud Solution Providers (CSP), and more.

- [Azure DevOps pricing](#)
- [Azure DevOps billing overview](#)

Billing for Microsoft Team Foundation Server (TFS)

You can also make some purchases for TFS on your monthly Azure bill.

- [TFS pricing](#)
- [How to buy TFS CALs or access to the TFS Test Services page](#)

5-Minute quickstarts

- [Set up billing for Azure DevOps](#)
- [Pay for Azure DevOps users](#)
- [Buy Azure DevOps Pipelines](#)
- [Start free trials for paid Azure DevOps features and extensions](#)
- [Buy cloud-based load testing](#)

Step-by-step tutorials

- [Connect your Azure DevOps organization to Azure Active Directory](#)

How-to guides

Billing management

- [Add user to make purchases or backup billing manager](#)
- [Change the Azure subscription your Azure DevOps organization uses for billing](#)
- [Change the number of paid users on your Azure DevOps organization](#)
- [Billing FAQ](#)

Marketplace extension management

- [Change the number of paid users for an Azure DevOps extension](#)
- [Approve requests for extensions](#)
- [Assign paid extensions to users](#)
- [Uninstall or disable extensions](#)

Guidance for Cloud Solution Providers

- [Set up your customer](#)

- [Buy Azure DevOps for your customers](#)
- [Buy and manage Visual Studio subscriptions](#)
- [Buy App Center resources](#)

Troubleshooting

- [Trials](#)

Reference

- [Permissions](#)
- [About access levels](#)
- [Default permissions & access](#)

Other resources

- [Start using Azure DevOps](#)
- [Manage organizations](#)
- [TFS Server Administration documentation](#)
- [Buy Visual Studio cloud subscriptions](#)
- [Buy Xamarin University subscriptions](#)