simplilearn

Getting Started with

Azure DevOps

TABLE OF CONTENTS

GETTING STARTED WITH AZURE DEVOPS

Azure DevOps gives DevOps teams the tools they need to plan work, share code, exchange development environments, and deploy applications. Developers can either use DevOps in the cloud using Azure DevOps Services or on-premises using Azure DevOps Server. You may know Azure DevOps Server by the name it used to go by: Visual Studio Team Foundation Server (VS TFS).

You can use Azure DevOps tools with your web browser or IDE client. Microsoft allows you to use the following services based on your development needs:

- **Azure Repos** offers Git repositories or Team Foundation Version Control (TFVC). These services manage for source control of your code.
- **Azure Pipelines** let you run build and release services with the result of supporting continuous integration and delivery.
- **Azure Boards** give you easy access to the Agile tools required planning, code defects, tracking work, and issues using Scrum and Kanban.
- **Azure Test Plans** are services that give tools to run manual, exploratory, and continuous testing.
- **Azure Artifacts** run Maven, npm, and NuGet packages from public and private sources and integrate package sharing into your CI/CD pipelines.

You are not limited to Microsoft's tools. In addition to those, you can use the following collaboration tools such as:

- Customizable team dashboards
- Configurable widgets in your dashboard to share information, progress, and trends
- Wikis for sharing information
- Notifications you can configure

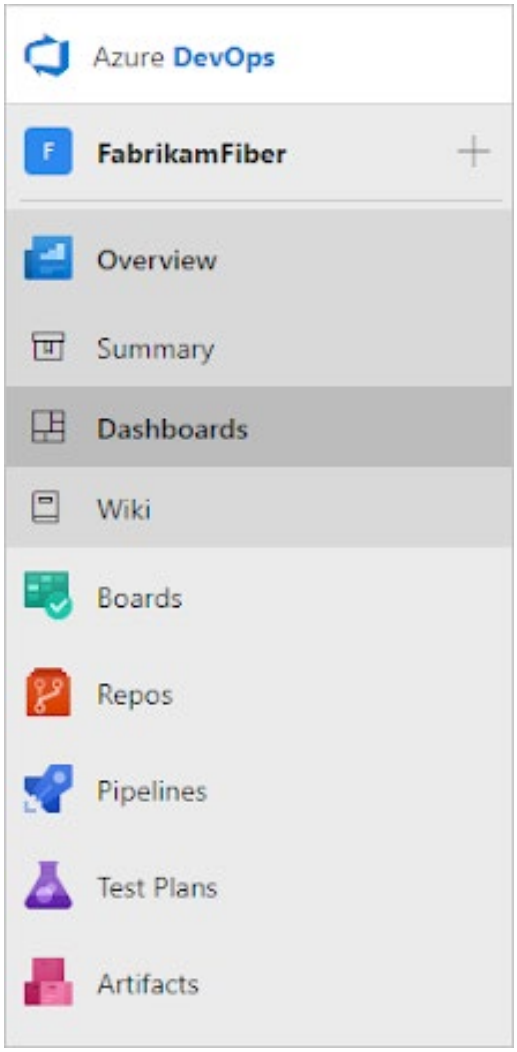
You will find extensions for integrating popular services, such as Campfire, Slack, Trello, UserVoice, and more and developing your custom extensions with Azure DevOps.

WHAT SERVICES AND FEATURES DO YOU GET WITH AZURE DEVOPS?

You get a connected set of services and tools to manage your software projects when using Azure DevOps. The tools range from planning and development through testing and deployment. You will find that each service is sent through a client/server model. Most of the services can be accessed using an easy-to-use web interface accessible through all major browsers that run on desktop, tablet, and phones. Dedicated clients are available for source control, build pipelines, and work tracking.

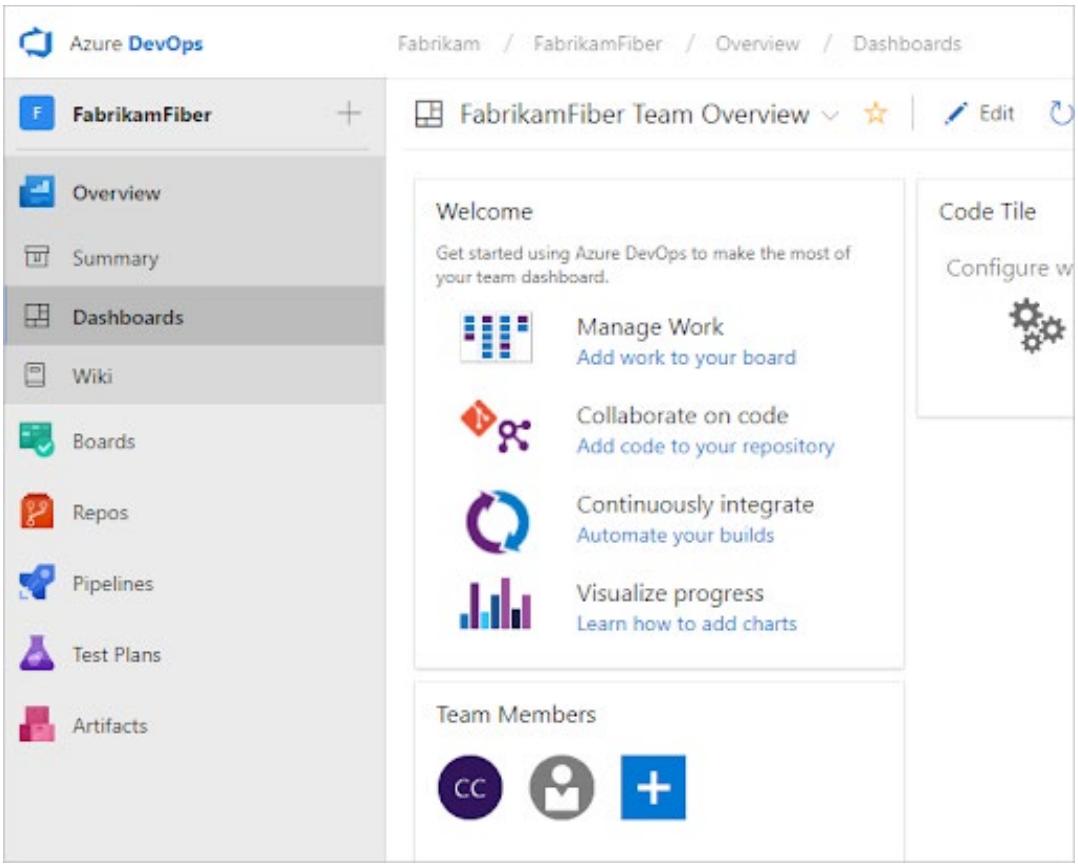
The following images show how to access web services.

Many of Azure's DevOps services are either free for small teams (ten or less) or available through a subscription or per-use model. The good news is that you can use paid and free services to manage the work you are doing. With Azure's DevOps services, you only pay for what you need when you need it.



Dashboards

The Dashboards give you access to user-configurable screens, or Dashboards, to manage your environment.



Dashboards allow you to do the following tasks:

- Add, configure, and manage dashboards
- Customize widgets that you add to dashboards
- Navigate to different areas of your project

Source control

Developers use source or version control systems to collaborate on projects and track changes made to the core codebase. Multi-developer projects require source control to manage the work the teams are doing.

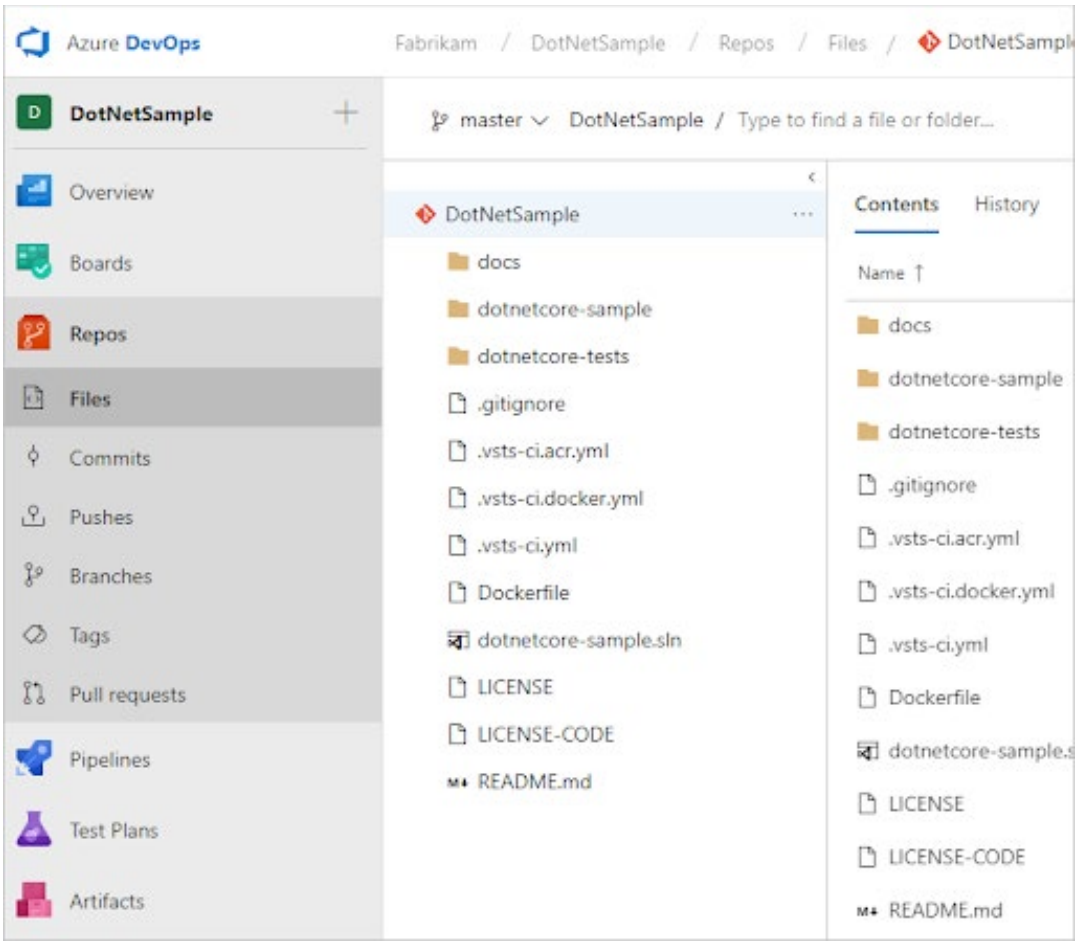
Azure DevOps systems support two types of source control:

- Git (distributed)
- Team Foundation Version Control (TFVC), is a centralized, client-server system

Both systems give you the same ability to check-in files and organize files within folders, branches, and repositories. Simplilearn has several videos on YouTube (**https://www.youtube.com/watch?v=hDxIdPxKt5w**) you can watch on how to use Git. The learnings can then be used with the Azure DevOps environment.

A developer has a copy on their dev machine of the source repository when they are using Git that also includes all branch and history information. Across your team, every developer works directly with their local pool, and changes are shared between repositories as a separate step.

Developers send each set of modifications and do version control operations like history and compare without a network connection. Each branch is intentionally lightweight. A developer creates a local private branch when they need to switch context and switch from one branch to another to move among different variations of the codebase. As a solution matures, developers can merge, publish, or dispose of the branch.



You can do the following tasks when you use Azure Repos for Git:

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

Plan and track work

It is essential for DevOps teams to easily access information and track the status of work, tasks, issues, or code defects. You may have used tools such as Microsoft Excel, Microsoft Project, possibly a bug tracking system, or a combination of instruments. Modern development teams have adopted Agile Agile methods and practices and the planning and development models Agile requires.

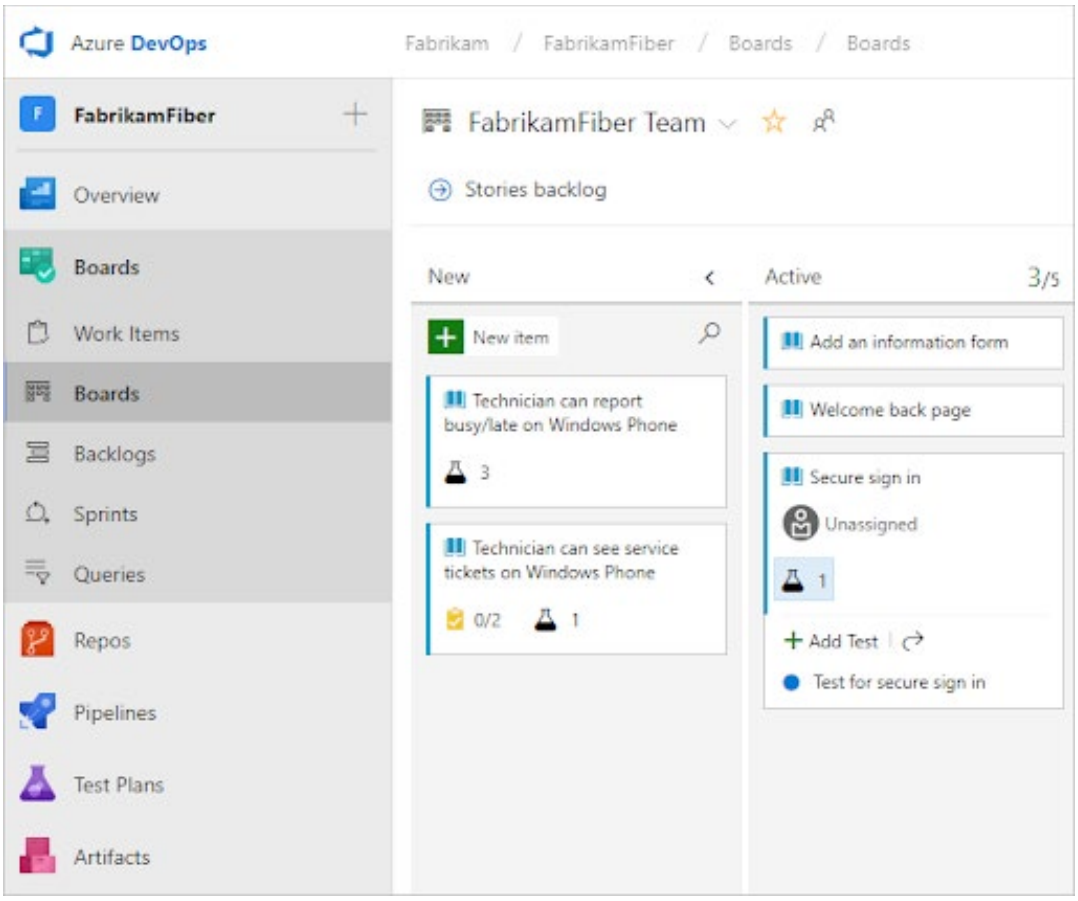
Azure DevOps systems offer different types of work items that you use to track features, requirements, user stories, tasks, bugs, and issues. You manage progress and work with work items associated with work item type and a set of fields that can be easily updated.

Scrum, Kanban, or Scrumban should be used for planning purposes. Azure gives you, and you have access to many types of backlogs and boards to support the main Agile methods.

- **Product backlog:** Use this to create and rank stories or requirements.
- **Kanban:** Use this to visualize and coordinate work as it progresses from beginning to in-progress, to done.
- **Sprint backlogs:** Use this to plan work to complete during a sprint cycle, a regular two to a four-week cadence that teams use when implementing Scrum.
- **Taskboard:** Used during daily Scrum meetings to review work that's completed, remaining, or blocked.

Information is tracked on the backlogs and boards that allow scrum leads and developers easy access to the current workload. Charts and dashboards complement the boards and provide a complete picture to visualize team progress and trends.

Below is a screenshot of how to access Boards. Boards are the tool you can use for Agile tools to support planning and tracking work.



Individually, you can complete the following tasks:

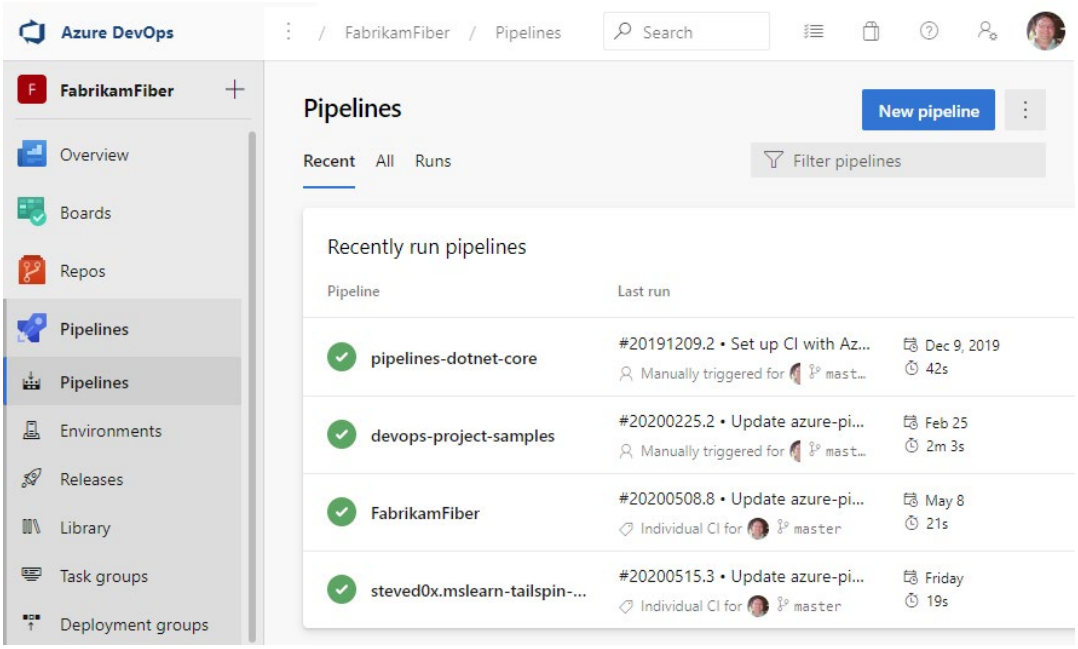
- Add and update work items
- Define work item queries
- Create status and trend charts
- Manage the product backlog
- Plan sprints by using sprint backlogs
- Review sprint tasks and update tasks through the task boards
- Visualize the workflow and update the status by using Kanban boards
- Manage portfolios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

Continuous integration and deployment

By automating as many processes as possible, you increase the pace and consistency of the software release. Azure systems provide you with support to build, test, and release automation.

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

Azure Pipelines offer a connected set of features to support building and deploying your applications.



You will use pipelines to implement continuous integration and continuous delivery of your software solutions.

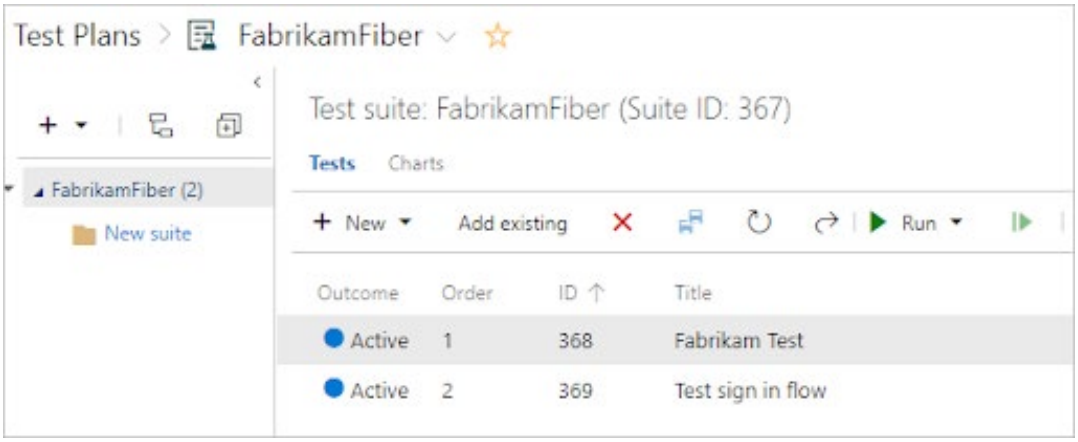
- **Build automation:** Outline the steps you need to take during the build and the events that start a build.

- **Release management:** Increase the number of release cadences and control of simultaneous releases. (Azure can manage the release pipelines that represent your environments, from development to production.) Run automation to move your app to each situation. Add approvers to validate that the app has been successfully deployed to an environment. Then create your release manually or automatically from a build. Finally, track your releases as they're deployed to various environments.

Manual and exploratory testing

Test features support exploratory and manual testing, along with continuous testing.

Test Plans support creating and managing manual tests.



You can leverage the following test features:

- Test plan, test suite, and test case work items you can use to customize workflows
- End-to-end traceability that covers requirements to test cases and bugs with requirements-based test suites
- Criteria-based test selection with query-based test suites
- An interface that looks like Excel with a grid for fast test cases
- Reusable test steps: you can share the test data with shared levels 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

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
- Ability to request and manage feedback
- Analytics service, analytic views, and Power BI reporting

Service hooks

Hosted on Azure DevOps, Service hooks actions complete tasks on other services when events happen within your project. An example is sending a push notification to a smartphone when a build fails. You can also use service hooks with custom apps and services. A Service hook is a more efficient way to drive activities in your projects.

AZURE DEVOPS SERVICES VS. AZURE DEVOPS SERVER

Azure DevOps Services and Azure DevOps Server were initially named Visual Studio Team Services (VSTS) and Team Foundation Server (TFS). TFS is a legacy on-premise solution that teams have been using for more than a decade.

Recently both Azure DevOps Services and Azure DevOps Server have started to offer an integrated environment that supports Git, continuous integration, and Agile tools.

There is a difference between the two tools. Azure DevOps Services is a cloud offering. VSTF is a scalable, reliable, and globally available hosted service. VSTF is backed by a 99.9% SLA, monitored by Microsoft's 24/7 operations team, and available in local data centers worldwide.

The other tool, Azure DevOps Server, is an on-premises package that uses SQL Server to manage all transactions developers will do. An on-premise solution means that the software was purchased and installed on a server managed in a private enterprise datacenter. Companies will select on-premises when they need their data to stay within their network.

Although both Azure DevOps Services and Azure DevOps Server provide the same foundational services, compared with Azure DevOps Server, Azure DevOps Services also includes the following :

- Easy to use server management.
- Real-time access to the latest features
- Improved connectivity with remote sites.
- A transition from capital expenditures (servers and the like) to operational expenses (subscriptions).

It is much easier for Microsoft to add new features to Azure DevOps Services. The updates are immediately available to all users. In contrast, Azure DevOps Server requires a team to install updates on the server.

Core differences between Azure DevOps Services and Azure DevOps Server

When you're choosing between Azure DevOps Services and Azure DevOps Server consider the following areas:

- Scope and scale data
- Authentication
- Users and groups
- Manage user access
- Security and data protection

There are pros and cons to both environments. With that said, it is clear that Microsoft is investing heavily in its cloud-hosted services. For this reason, if your team is choosing between the two, you may want to think about the long term investment Microsoft is making with the Azure cloud services.

Scope and scale data

When you purchase Azure DevOps Services, you have two choices for scoping and scaling data: organizations and projects. If you are an organization using Azure DevOps Services, you get your URLs (for example, https://dev.azure.com/fabrikamfiber), and you always have exactly one project collection. You will be able to manage a large number of projects within a group.

If you select a project for your scaling, then you will only have access to the one project you create.

Scaling Azure DevOps Servers with deployments, project collections, and projects

Azure DevOps Server offers the following configurations for scoping and scaling data:

- Deployments
- Project collections
- Projects

In many ways, you can think of deployments as just servers. With that said, implementations can be more complicated. You should consider the following:

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

Project collections become containers for security and administration. Also, each collection is a physical database boundary. They're also used to group related projects.

Projects are used to manage the assets of individual software projects, including source code, work items, and so on.

Authentication

You connect over the public internet when you are using Azure DevOps Services. For example, the web address you would connect to would look something like https://simplilearn.visualstudio.com. You either validate your credentials with a Microsoft account or Azure AD.

Microsofts suggests you use Azure AD rather than Microsoft accounts to validate the account. You have enhanced security when using Azure AD. For instance, when an employee leaves the company, and their Azure AD account is deactivated, their access to Azure DevOps Services is automatically revoked.

Access to an intranet server with Azure DevOps Services is possible. You authenticate with Windows Authentication and your Active Directory (AD) domain credentials. This process is managed with AMS (Access Management Services) to create a transparent sign-in experience.

Manage users and groups

You can use similar control with Azure DevOps Services to offer access to groups of users. You can add Azure AD groups to Azure DevOps Services groups. If you use Microsoft Accounts instead of Azure AD, you have to add users one at a time.

In Azure DevOps, Server leverages Active Directory (AD) groups to give users access to deployments. You will create Azure DevOps groups in Active Directory. The Active Directory group memberships are kept in sync. You gain and lose access to Azure DevOps Server as you are added and removed in Active Directory.

Manage user access

You control access in both Azure DevOps Services and Azure DevOps Server to specific features by assigning users to an access level. You must assign each user to a single access level. With both Azure DevOps Services and Azure DevOps Server, you can offer 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. Microsoft only charges for other users who need access.

You must give access to Azure DevOps Services for each user level in your organization. Visual Studio subscribers are validated by Azure DevOps Services as they sign-in. Azure DevOps Services allows you to grant Basic access for up to five users without Visual Studio subscriptions. If you require more than five people to have Basic access, you will need to pay for the additional accounts. Also, you can give the other account Stakeholder access.

You can give groups of user access with Azure AD groups. The appropriate access is granted when you first sign-in.

The honor system applies to Azure DevOps Server. All users have Basic access Azure DevOps Server/TFS Client Access License (CAL). Depending on your Visual Studio subscription, Azure DevOps Server determines if you get Basic or Advanced access. Azure DevOps Server cannot validate licenses or enforce compliance.

Security and data protection

Many company's first concern when moving to the cloud is security. Will my projects be secure if I am using Azure DevOps Services? The short answer is: Yes. Microsoft is committed to having security at the center of its cloud services. Microsoft's commitment covers Azure, Micorosft 365, and the many additional services they run in the cloud. The same central principle is evident with Azure DevOps Services.

Process customization

Depending on which product you buy (Azure DevOps Services or Azure DevOps Server), you will have different ways to customize the work-tracking experiences. The following process models are supported:

- **Azure DevOps Services:** leverages the Inheritance process model, which supports WYSIWYG customization
- **Azure DevOps Server:** you can choose the Inheritance process model or the On-premises XML process model, which promotes customization through import or export of XML definition files for work-tracking objects
- **Azure DevOps Server 2018 and earlier versions:** you only have access to the On-premises XML process model

Azure DevOps Server does support an XML process model, and it is a compelling option. There is, however, one central issue: processes for existing projects will not be automatically updated.

A good example is how Azure DevOps Server 2013 introduced many new features that require new work-item types and other process template changes. When you upgrade from 2012 to 2013, these changes will not be automatically incorporated into your existing projects. Unfortunately, after you have completed the upgrade process, you will have to use the Configure features wizard and new manual processes to align each project.

TAKING IT TO THE NEXT LEVEL

The focus of this book is to introduce Azure's DevOps tools. The tools themselves build on top of the DevOps model that is being adopted by so many companies. Microsoft is committed to DevOps, DevSecOps, SRE, and all future development models. Expect more to come from Microsoft and the Azure team.

INDIA

Simplilearn Solutions Pvt Ltd.

53/1 C, Manoj Arcade, 24th Main,
Harlkunte
2nd Sector, HSR Layout
Bangalore - 560102

Call us at: 1800-212-7688

USA

Simplilearn Americas, Inc.

201 Spear Street, Suite 1100,
San Francisco, CA 94105
United States

Phone No: +1-844-532-7688

www.simplilearn.com