Azure DevOps (TFS/VSTS) vs GitLab

GitLab compared to other DevOps tools

On this page

Summary

On September 10, 2018 Microsoft renamed VSTS to Azure DevOps and TFS to Azure DevOps Server. and upgraded both with the same new user interface.

Azure DevOps (VSTS) is a hosted cloud offering, and Azure DevOps Server (TFS), is an on-premises version. Both offer functionality that cover multiple stages of the DevOps lifecycle including planning tools, source code managment (SCM), and CI/CD.

As part of their SCM functionality, both platforms offer two methods of version control.

1. Git (distributed) - each developer has a copy on their dev machine of the source repository including all branch and history information.
2. Team Foundation Version Control (TFVC), a centralized, client-server system - developers have only one version of each file on their dev machines. Historical data is maintained only on the server.

Microsoft recommends customers use Git for version control unless there is a specific need for centralized version control features.
https://docs.microsoft.com/en-us/vsts/tfvc/comparison-git-tfvc

This is noteworthy given that in June of 2018 Microsoft purchased GitHub, the Internets largest online code repository.

Resources

- Azure DevOps
- Azure DevOps Announcement Blog
- Azure DevOps public Roadmap and Release History
- Visual Studio Team Services
- Team Foundation Server)

Comments/Anecdotes

- Lots of emphasis on cross platform (windows, Mac, Linux), and free macOS CI/CD is pretty rare.
- From https://azure.microsoft.com/en-us/blog/introducing-azure-devops/

  > Azure DevOps represents the evolution of Visual Studio Team Services (VSTS). VSTS users will be upgraded into Azure DevOps projects automatically. For existing users, there is no loss of functionally, simply more choice and control. The end to end traceability and integration that has been the hallmark of VSTS is all there. Azure DevOps services work great together.

  > As part of this change, the services have an updated user experience.

> Users of the on-premises Team Foundation Server (TFS) will continue to receive updates based on features live in Azure DevOps. Starting with next version of TFS, the product will be called Azure DevOps Server and will continue to be enhanced through our normal cadence of updates.

- HackerNews comments saying it's just a rebrand - PM for AzureDevOps responding:

  > PM for Azure DevOps here (formerly VSTS). It is a rebranding, but it's more than merely a rebranding. We're breaking out the individual services so that they're easier to adopt. For example, if you're just interested in pipelines, you can adopt only pipelines.

- From a call with a prospect Bank:
  - Went with Azure DevOps because

    > It's platform agnostic, it's in the cloud, great capabilityality, tons of functionality, it does what we need it to do. We like it a lot. It really has nothing to do with Microsoft. Microsoft is very agnostic and open source embracing now, so that the old Java vs .Net thing is kind of over.

  - Appealed to a shop that was "more Java than Microsoft technologies". But they had lots of the Microsoft development suite already, and trusted where Microsoft is going.
  - Azure DevOps is dropping new releases every sprint (2-3 weeks). Their roadmap is public:Azure DevOps public Roadmap and Release History

Pricing

Azure DevOps

Azure DevOps Services Pricing
Azure Pipelines Only Pricing
Azure DevOps On-prem = See TFS Pricing

VSTS

VSTS Pricing

Visual Studio â€˜Professional Versionâ€™ is the most comparable to GitLab since Visual Studio â€˜Enterprise Versionâ€™ includes extras outside the scope of DevOps (such as MS Office, etc).

Visual Studio Professional can be purchased under a â€˜standardâ€™ or â€˜cloudâ€™ model.

- Standard = $1,200 year one (retail pricing), then $800 annual renewals (retail pricing)
- Cloud - $540 per year

Under their â€˜modern purchasing modelâ€™, the monthly cost for Visual Studio Professional (which includes TFS and CAL license) is $45 / mo ($540 / yr). Â However, extensions to TFS such as Test Manager ($52/mo), Package Management ($15/mo), and Private Pipelines ($15/mo) require an additional purchase.

TFS

TFS Pricing

A TFS license can be purchased as standalone product, but a TFS license (and CAL license) is also included when you buy a Visual Studio license / subscription.

MS pushes Visual Studio subscriptions and refers customers who are only interested in a standalone TFS with a â€˜classic purchasingâ€™ model to license from a reseller.

Excluding CapEx and Windows operating system license, a standalone TFS license through a reseller in classic purchasing model is approximately $225 per year per instance. The approximate Client Access License is approximately $320 per year.

Comparison

FEATURES		
Commit graph and reporting tools GitLab provides commit graphs and reporting tools about collaborators' work. Learn more about commit graphs	✓	✓
The most comprehensive import feature set GitLab can import projects and issues from more sources (GitHub, BitBucket, Google Code, FogBugz, Gitea and from any git URL) than GitHub or any other VCS. We even have you covered for your move from SVN to Git with comprehensive guides and documentation. Making it easier to get up and running with GitLab	✓	✓
Wiki based project documentation A separate system for documentation called Wiki, is built right into each GitLab project. Every Wiki is a separate Git repository. Learn more about GitLab Wikis	✓	✓
Image Discussions Within a commit view or a merge request diff view, and with respect to a specific location of an image, you can have a resolvable discussion. Have multiple discussions specifying different areas of an image. Image Discussions	✗	✓
Merge Request Commit Discussions Comment on a commit within the context of a merge request itself Merge Request Commit Discussions	−	✓
Create merge request from email Create a merge request from email by sending in the merge request title, description, and source branch name. Create merge request from email	✗	✓

Lock Discussion

Lock down continued discussion in an issue or merge request as a Master role or higher, to prevent further abuse, spam, or unproductive collaboration.

Lock Discussion

✗ ✓

First time contributor badge

Highlight first-time contributors in a project.

✗ ✓

Preview your changes with Review Apps

With GitLab CI/CD you can create a new environment for each one of your branches, speeding up your development process. Spin up dynamic environments for your merge requests with the ability to preview your branch in a live environment.

Learn more about Review Apps

⊖ ✓

Multiple approvals in code review

In GitLab, to ensure strict code review, you can require a specific number of approvals on a merge request by different users before being able to merge it. You can undo an approval by removing it after the fact.

Approvals Documentation

✓ ✓

Ease of migration from other providers

GitLab lets you easily migrate all repos, issues and merge request data from your previous provider.

Learn how to migrate your projects to GitLab

✗ ✓

Search files with fuzzy file finder

GitLab provides a way to search a file in your repository in one keystroke.

Read about the file finder in our documentation

✓ ✓

Fast-forward merge with option to rebase

With this setting at the project level, you can ensure that no merge commits are created and all merges are fast-forwarded. When a fast-forward merge is not possible, the user is given the option to rebase.

Learn more about rebase before merge

✓ ✓

Squash and merge Combine commits into one so that main branch has a simpler to follow and revert history. Learn more about squash and merge	✓	✓
Import from GitLab.com Import projects from GitLab.com to a private GitLab instance. Learn more about importing projects from GitLab.com	✓	✓
Limit project size at a global, group, and project level Ensure that disk space usage is under control. Learn more about project size limiting	✗	✓
Merge request approvals When a project requires multiple sign-offs, GitLab Enterprise Edition enables you to make sure every merge request is approved by one or more people. Merge request approvals allow you to set the number of necessary approvals and predefine a list of approvers that will need to approve every merge request in a project, and in-turn improve your codeâ€™s quality. Learn more about merge request approvals	✓	✓
Merge Requests Create merge requests and @mention team members to review and safely merge your changes. Learn more about merge requests	✓	✓
Merge conflict resolution Preview merge conflicts in the GitLab UI and tell Git which version to use. Learn more about the merge conflict resolution UI	✓	✓
Reject unsigned commits GitLab Enterprise Edition Premium allows you to enforce GPG signatures by rejecting unsigned commits. Read more about enforcing push rules	✗	✓

Verified Committer Verify that a push only contains commits by the same user performing the push. In development for GitLab. Follow this link for more information.	✗	✓
Git has smaller size requirements A single repository in Git is typically a number of times smaller than the same repository in SVN. Read about using Git with GitLab	✓	✓
File Locking Working with multiple people on the same file can be a risk. Conflicts when merging a non-text file are hard to overcome and will require a lot of manual work to resolve. With GitLab Enterprise Edition Premium, File Locking helps you avoid merge conflicts and better manage your binary files by preventing everyone, except you, from modifying a specific file or entire directory. Learn more about File Locking	✓	✓
Merge when pipeline succeeds When reviewing a merge request that looks ready to merge but still has one or more CI/CD jobs running, you can set it to be merged automatically when the jobs pipeline succeeds. Learn more about Merge when pipeline succeeds	✓	✓
Revert specific commits or a merge request from the UI Revert any commit or a single merge request from GitLab's UI, with a click of a button. Learn how to revert a commit or a merge request from the GitLab UI.	✓	✓
Powerful branching A branch in Git contains the entire history that preceeds it. It's also created or moved towards instantly and easily shared. See the Git documentation to get started with branches	✓	✓
Protected branches Granular permissions for branches you want to protect. Read about protected branches	✓	✓

Web IDE

Contribute to projects faster by using the Web IDE to avoid context switching in your local development environement. The Web IDE is integrated with merge requests and GitLab CI so that you can resolve feedback, fix failing tests and preview changes live with client side evaluation without leaing the Web IDE.

Learn more about the Web IDE

✗ ✓

Snippets

Store and share code snippets to engage in a conversation about that piece of code. You can embed snippets on any blog or website using a single line of code.

Learn more about Snippets

✓ ✓

Merge request versions

View and compare merge request diffs from the merge request UI.

Learn more about merge request versions

✗ ✓

Inline commenting and discussion resolution

Code or text review is faster and more effective with inline comments in merge requests. Leave comments and resolve discussions on specific lines of code. In GitLab, Merge Request inline comments are interpreted as a discussion. You can configure your project to only accept merge requests when all discussions are resolved.

Learn more about resolving discussions

✓ ✓

Cherry-picking changes

Cherry-pick any commit in the UI by simply clicking the Cherry-Pick button in a merged merge request or a specific commit.

Learn more about cherry picking merge requests

✓ ✓

Activity Stream

View a list of the latest commits, merges, comments, and team members on your project.

Learn more about the Activity Stream

✓ ✓

Custom Notifications

Be notified by email, Slack, or ToDos anytime there are changes to an issue or merge request.

Learn more about Custom Notifications

✓ ✓

GPG Signed Commits Sign commits and prove that a commit was performed by a certain user. Read more about GPG signed commits	✗	✓
Work in Progress merge requests (WIP) Prevent merge requests from accidentally being accepted before they're completely ready by marking them as Work In Progress (WIP). This gives you all the code review power of merge requests, while protecting unfinished work. Learn more about WIP MRs	⊖	✓
Restrict push and merge access to certain users Extend the base functionality of protected branches and choose which users can push or merge to a protected branch. Read about protected branches	✓	✓
Protected tags Granular permissions for tags you want to protect. Read about protected tags	✗	✓
Built-in and custom project templates When creating a new project, you can choose to kickstart your project from a predefined template that already has some working example code and CI preconfigured. In addition, you can define a custom project templates by assigning a group. Child projects of this group are available as templates when creating a new project. Read more about Project templates	✗	✓
Create projects with Git push Push new projects to the desired location and a new private project will automatically be created. Learn more about creating Projects	✗	✓