**What is VSTS(Azure DevOps) and Difference between TFS and VSTS, GIT.**

**TFS**: All bugs, tasks, eng. Stories were logged in TFS. But there was no dashboard to trace out the graphically result for team. So every dev, lead had to be expert in excel to track the work items more effectively.

**VSO\VSTS**

In 2013, Microsoft introduced VSO (Visual studio online). At the time of launch **VSO appeared to be nothing more than TFS on cloud**. However, MS started adding more and more features to VSO. Many of these features were inspired from competitive tools. Over a period of time, **VSO was aptly renamed to VSTS (Visual Studio Team Service)**. Today it is one stop for all our development.

Today its name also rename with Azure DevOps. Azure DevOps(VSTS) has solved one big problem of fragmentation. Its features today on -par

VSTS = Git/TFS + JIRA + Team City + Octopus.

Now, its offer full support for more and more non-Microsoft services. If you don’t want to use VSTS for everything, you still have a choice to choose what you want. Fore example, you can choose octopus over VSTS Release management and push package from VSTS build to octopus directly.

**VSTS VS TFS**

1. VSTS is online tool and configured on cloud services

TFS (TVFC) is avail in On-premises form

1. VSTS can be accessible anywhere by using internet with Microsoft account.

TFS can be accessible only within our premises.

1. User don’t need to maintain or upgrade any server for VSTS which is configured on cloud.

Timely server maintenance or upgrade is required for TFS.

1. No hardware/OS upgradation is required

Hardware/OS upgradation is required for TFS

1. VSTS has SharePoint, Power BI, Excel, MS Teams, Office 360, integration features

NA for TFS

1. Load Testing can be done on cloud using VSTS

TFS Can’t be done load testing.

1. Dashboard functionality for Scrum master and graphically output as per work item progress.

TFS Only Work item list can get through query.

1. It has all functionality of TFS such as Source code management, build, release handling, shelveset , code review, merging , version control

TFS Handle Application life cycle management, mange the work items, source code, version control, build, reports, Shelveset,

1. VSTS, you connect with a public internet, either with MS Account or Azure AD.

TFS typically connect to an internet server and authenticate with windows AD credentials.

1. VSTS only separates into two options for scoping and scaling data: accounts and team projects.

TFS has three options for it. Deployment, team project collections and team projects.

**TFVS VS Git.**

Team Foundation Version Control is a centralized version control system. Only one version of each file on their dev machine. Historical data is maintained only on the server. Branches are path-bases and created on the server.

Git is a distributed version control system. Each developer has a copy of the Source repository on their dev machine. Developers can commit each set of changes on their dev machine and perform version control operations such as history and compare without a network connection. Branches are light weight. You can create a private local branch. Later you can merge , publish or dispose of the branch. This is third party service.

It has two workflow models: Server Workspaces and Local Workspaces. Developers usually work in Local workspaces workflow model. In each team member takes a copy of the latest version of the codebase with them and works offline as needed. They check-in their changes and resolve conflicts as necessary.

You might have to resolve conflicts when you get check in, merge or unshelve.

1. Git You might have to resolve conflicts when you pull or merge. You can resolve content conflicts in VS or from the command prompt.

You can view history in visual studio and on the web portal (VSTS).

2. TFS has a great feature called “Shelve-Sets”. This allows one to take pending changes and shelve them in order to work on something different and not affect the current state of the repository.

Git has a similar feature called stashing, or “git stash” using proper parlance. The biggest difference between the two is that stashing is only done at your local repository. You cannot “push” the stash (or stashes – you can have multiples) up to the shared remote.

‘

**Microsoft Visual Studio Team Services (VSTS) Tutorial: The Cloud ALM Platform**

Microsoft Visual Studio Team Services (VSTS) is a new cloud-based offering that helps the project teams to take care of all the aspects of the process for software development.

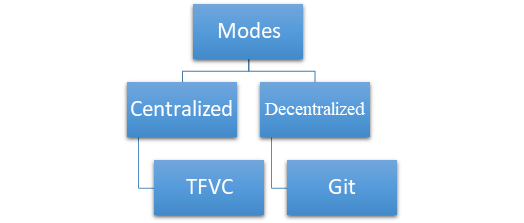
Visual Studio Team Services (VSTS) is an online hosted service from Microsoft

Microsoft VSTS is an **Application Lifecycle Management (ALM)** system which helps the entire project team to capture Requirements, Agile /Traditional Project Planning, Work Item management, Version Control, Build, Deployment, and manual Testing all in a single platform.

**In simple terms, Microsoft VSTS is Team Foundation Server (TFS) on the cloud.**

VSTS is tightly integrated with Visual Studio. NET IDE.

**Two Modes OF VSTS - Centralized and Decentralized**

****

A traditional model is a centralized model, where code created by many developers is stored at the server. This mode is also called Team Foundation Version Control.

Another mode is the decentralized model, where each developer keeps their own repository of code in local repository first and merges those to the remote repository when required. This mode is also called Git.

We can set one mode as the default mode for our team project. In this article below, we will see how the decentralized mode such as Git works.

Before that, now let see the concept called “TFS” and what it actually is and how it is different from our VSTS.

**Brief of TFS**

* Team Foundation Server (commonly abbreviated to TFS) is a [Microsoft](https://en.wikipedia.org/wiki/Microsoft) On-Premises product similar to the cloud service VSTS.
* In simple words, TFS will work offline whereas Vsts will work online
* It also provides [source code management](https://en.wikipedia.org/wiki/Revision_control) in two modes (either with Team Foundation Version Control or [Git](https://en.wikipedia.org/wiki/Git_(software))) like vsts.

**Difference between VSTS and TFS**

|  |  |
| --- | --- |
| **VSTS** | **TFS** |
| It supports online | It supports offline |
| Just like a Rental Service | Buy an Entire Product |
| No Initial Setup and Maintenance is required by the buyer/user | All the Initial setup and maintenance needs to done by the buyer/user. |

**Similarities**

VSTS is an online version of TFS.

Both support the two modes TFVC and Git

**Main Advantage of TFS over VSTS**

* You can choose TFS when you need your data to stay within your network or you want access to SQL Server reporting services that integrate with TFS data and tools.

**Main Advantages of VSTS over TFS**

* Visual Studio Team Services requires no setup. A user can simply [sign in](https://en.wikipedia.org/wiki/Sign_in)using a [Microsoft account](https://en.wikipedia.org/wiki/Microsoft_account) to set up an environment, create projects and add team members with less effort.
* VSTS provides optimal hardware so that our infrastructure cost is avoided.
* Since the VSTS services are available on the internet, they can be easily used by teams that are distributed across multiple locations, without having any additional cost of VPN and overheads of security.

**Pricing For VSTS**

We are lucky if our team size is 5 because great service like VSTS is absolutely free up to 5 users.