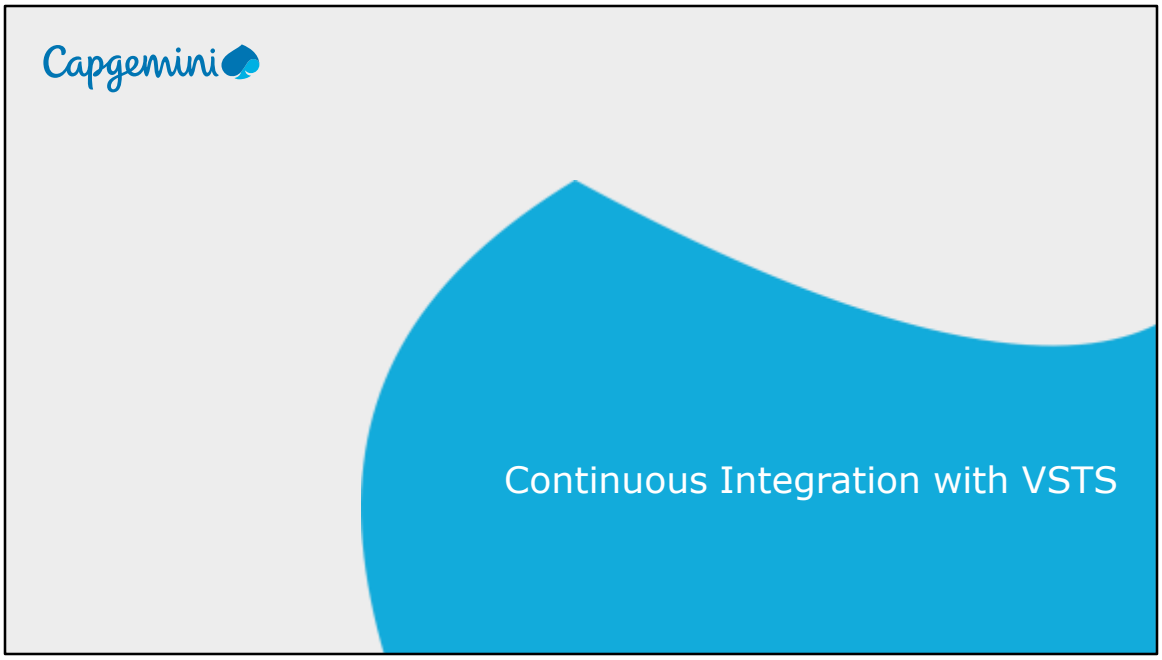

Capgemini
Continuous Integration with VSTS

Lesson Objectives

At the end of this module you will be able to:

- ✓ Understand the key concepts of version control
- ✓ Understand the VSTS fundamentals
- ✓ Create a Team Services project
- ✓ Build and deploy the project using VSTS

Introduction to VSTS

- Visual Studio Team Services (VSTS) is a cloud service for collaborating on code development
- Based on Team Foundation Server(TFS)
- VSTS basically TFS in the cloud
- VSTS is TFS++.
- Microsoft is developing VSTS using VSTS

Team Foundation Server 2005

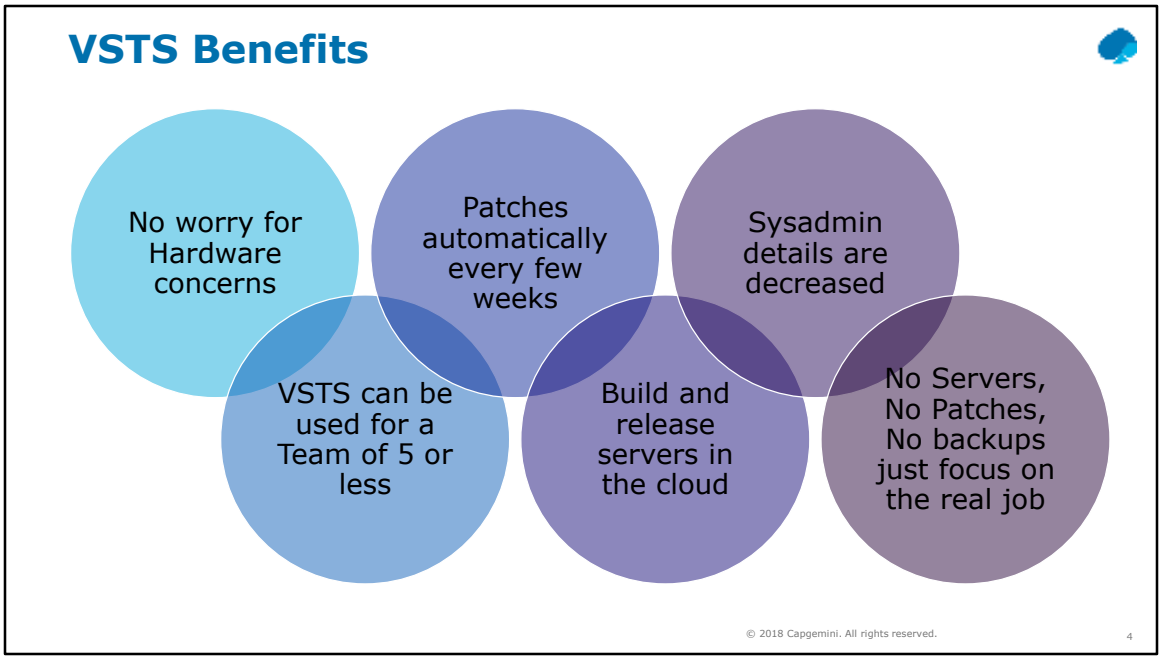
Team Foundation Server (TFS) had a lot of stuff right from the very beginning. It had source control, which was Team Foundation Source Control, (TFVC). It had automated build, project management tools, bugs and defect tracking tools. It had some basic dashboards and reports that were allowed on SharePoint and SQL Server Reporting Services. And it needed Visual Studio.

Team Foundation Server 2018

Team Foundation Server 2018 is the current version of on-premises TFS. It's got source control, and it's got the same source control that TFS 2005 had i.e. Team Foundation Version Control (TFVC) and It also supports Git. It has QA testing and defect tracking tools. So that we can manage and run manual test cases, which we used to do in the web. It also have project management tools, which supports Agile , Scrum, Kanban, Waterfall, CMMI or whatever you happen to be using, you can use TFS to manage your project. Dashboards don't rely on SQL Server Reporting Services or SharePoint anymore. Everything is being move to the web is great because TFS 2018 is super cross-platform we can run almost everything out of the browser. It supports Visual Studio Code, which exists on three different operating systems, so Windows, Mac OS, and Linux. TFS 2018 has a plugin for Eclipse, IntelliJ, Android Studio, Xcode, Xamarin, cross-platform command line client stuff. It works with Jenkins. TFS 2018 has good support to Git

VSTS it's not packaged software because nothing's actually getting shipped. Microsoft is doing everything in-house, which lets them deliver new features and new bug fixes

every few weeks



VSTS provides an integrated set of features that you access through your web browser or IDE client, including:

- 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

Version Control

- Version control is the integration point for everything you do
- Version control gives you tools for integrating your stuff, bringing all your code together.
- It gives you features for tracking and comparing history

VSTS Version Control Systems

Team Foundation Version Control (TFVC)

Git

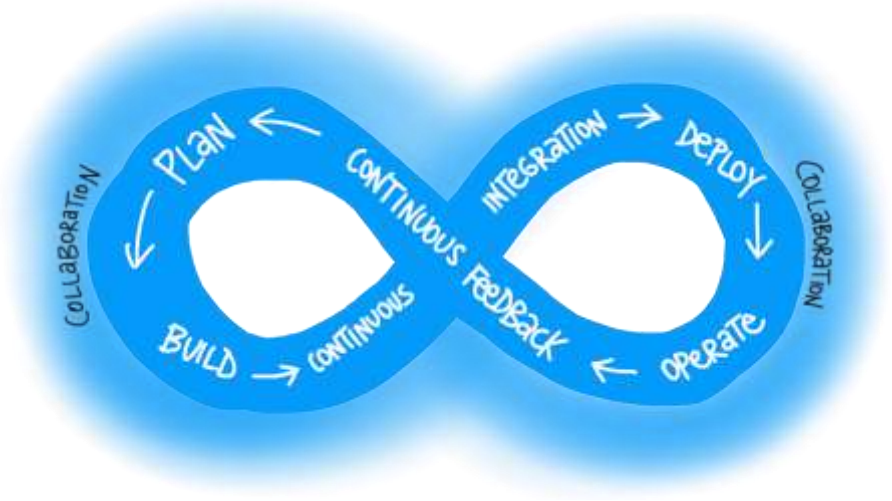
There are two types of version control systems available in VSTS

- **Team Foundation Version Control(TFVC)**
  - TFVC was created by Microsoft
  - It's been available in the product, and it's been available in TFS since the very beginning, all the way back starting with TFS 2005
  - TFVC is centralized version control
- **Git**
  - Git was created by Linus Torvalds
  - The initial release was back in April 2005. And support for Git was added to TFS and VSTS in 2013
  - Git is distributed version control

Centralized vs Distribute

Centralized Version Control	Distributed Version Control
TFVC is called centralized version control	Git is known as distributed.
Add, edit, delete, check in/out, branch, merge needs a network connection	Add, edit, delete, check in/out, branch, merge, undo available offline
Local workspaces give you the ability to do one level of undo.	Can be revert to any version
TFS knows what's on your local machine Changes are really easily tracked because you're connecting to the server pretty frequently	Remote server is completely unaware until you push to the remote repository
TFVC gives you really granular control over your security	Security is at the repository level.

DevOps



DevOps is the union of people, process, and products to create continuous delivery of value to end users. The contraction of "Dev" and "Ops" refers to replacing siloed development and operations teams with multidisciplinary teams that work together by using shared and efficient practices and tools. Essential DevOps practices include continuous integration and continuous delivery.

Continuous Integration

- Continuous integration(CI) is the practice that development teams use to automate merging and testing code
  - Build service in Team Services helps you set up and manage CI for your applications
- CI helps your team detect bugs early in the development cycle
  - Early detection makes bugs less expensive to fix.
  - Automated tests execute as part of the CI process to ensure quality
- Artifacts are produced from CI systems and fed to release pipelines to drive frequent deployments.

- Continuous integration(CI) is the practice in software engineering of merging all developer working copies several times to a shared mainline
- Continuous integration build is the fact that we build the integrated changes of our mainline by the means of a build server after each commit
- Commit changes to the source control system as often as possible
- Merge all these changes of all the developers into one main source control branch

Continuous Delivery

- Continuous delivery (CD) is a process by which code is built, tested, and deployed to one or more test and production environments.
  - Release service in Team Services helps you set up and manage CD for your applications
  - Deploying and testing in multiple environments drives quality
- CI systems produce deployable artifacts, including infrastructure and apps. Automated release pipelines consume these artifacts to release new versions and fixes to existing systems. Monitoring and alerting systems run continually to drive visibility in the entire CD process.

Team Services

- Team Services is a collection of hosted DevOps services for application developers
- **Build** and **Release** are two of the DevOps services in Team Services that help you manage continuous integration and delivery of your applications.
- Team Services has the following core components:
  - Code
  - Agents and queues
  - Builds
  - Release
  - Deployment groups

Code: Team Services hosts source-control repositories in which you can commit and push code, track a release across branches, and work toward release milestones

Agents and queues: To build your code or to deploy your software, you need at least one agent. An agent is installable software that runs one build or deployment job at a time.
Team Services supports the following agents:

- **Hosted agents** run on the Team Services platform. Microsoft takes care of the maintenance and upgrade of hosted agents.
- **Private agents** are agents that you set up and manage on your own systems to run build and deployment jobs

Builds: At the beginning of the build process, the agent downloads files from your remote repository to a local sources directory. You specify the events that will trigger the build, such as a code commit to a Team Services code repository.

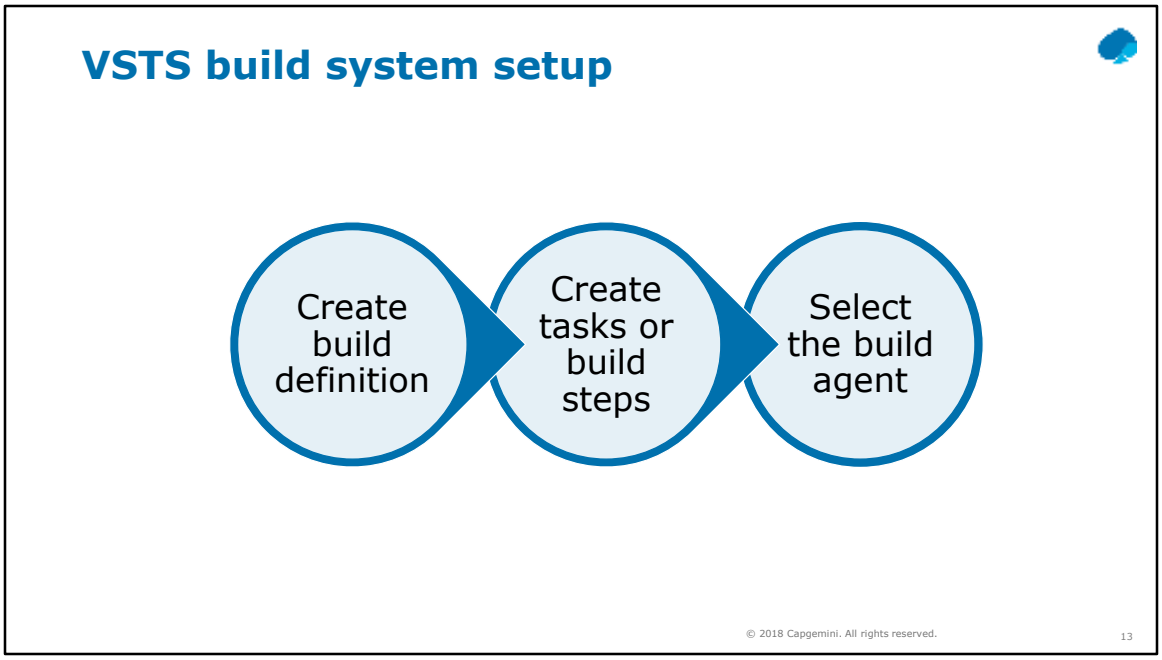
Release: A release definition is one of the fundamental concepts in Release Management for Team Services and Team Foundation Services. It defines the end-to-end release process for an application to deploy across various environments.

Deployment groups: A deployment group is a logical set of deployment target machines that have agents installed on each machine. Deployment groups represent the physical environments, such as "Dev," "Test," "UAT," and "Production." A deployment group is essentially another grouping of agents, much like an agent pool

Build Fundamentals

- A build system is a system used to automatically produce all the artifacts required to successfully deploy, test, and run our production software
- A system that can automatically trigger the builds and it produces all artifacts to successfully deploy, test, and run our production software
  - Automatically triggers the build when we commit new code to the source control system. No manual steps are involved in this process
  - Able to validate the software we build before we move into production
  - Deployments are also defined in some form of code and used during deployment.

Depending on the platform you're building for, you use various scripting technologies and templating languages. If you're working on Azure for example, you're using Azure Resource Manager templates. All these templates are coming from a source control system and the build is responsible for putting them in a place where we can get them from as one cohesive set of artifacts we use to put something into an environment.



Build definition

The build definition defines the steps that need to be executed in order for us to get the required artifacts we can deploy, test, and run our code in production. This build definition is more or less a template that is used to run our build over and over again.

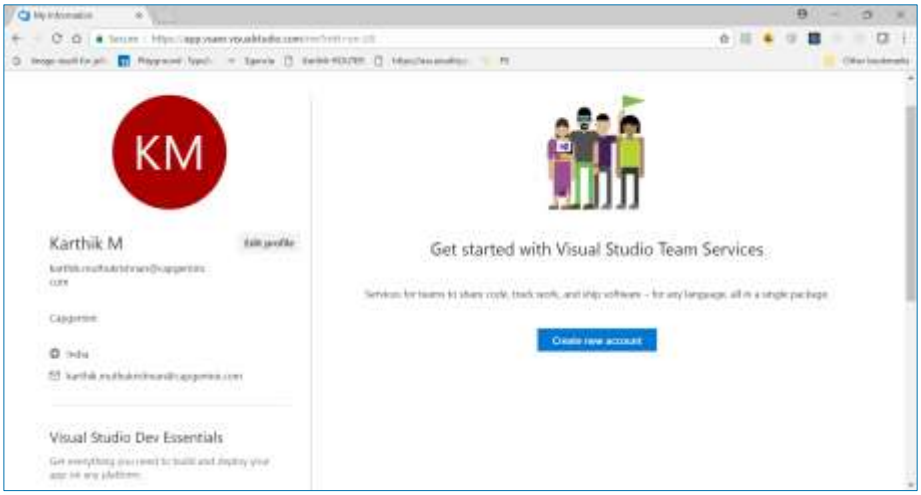
Build Steps

A task or build step is some form of automation that takes an input and produces a new output that is useful for us. For example, we can have a task that compiles the source code. It takes the input, what sources to build and where to find them, and it produces a result like for example a set of DLLs or executables. The build definition defines the sequence in which we run the tasks and it defines variables that we can pass along between the different tasks.

Build Agent

The build agent executes the set of tasks for us and does the actual work. So the build definition and the tasks define what the build should do. The agent takes this information and starts executing the tasks in the sequence defined in the template and then produces the actual end result.

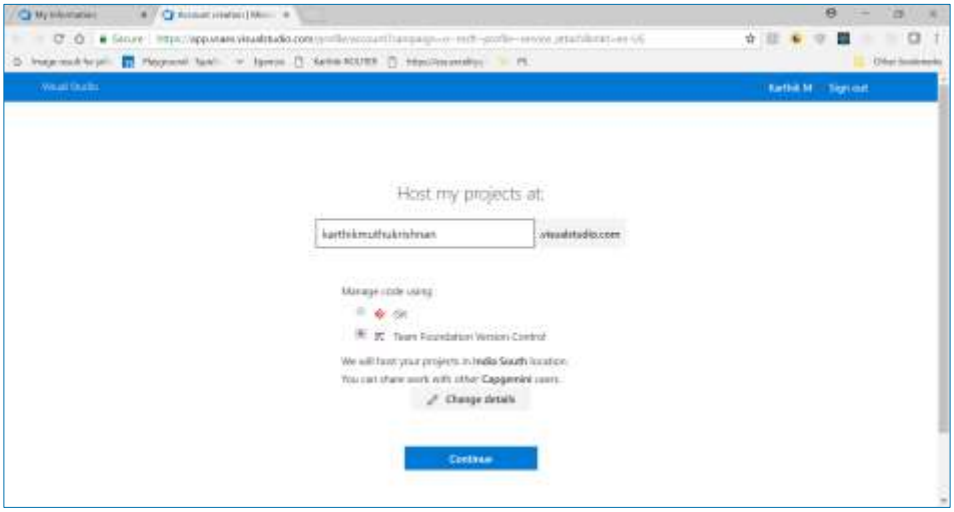
Getting started with VSTS – Creating Account



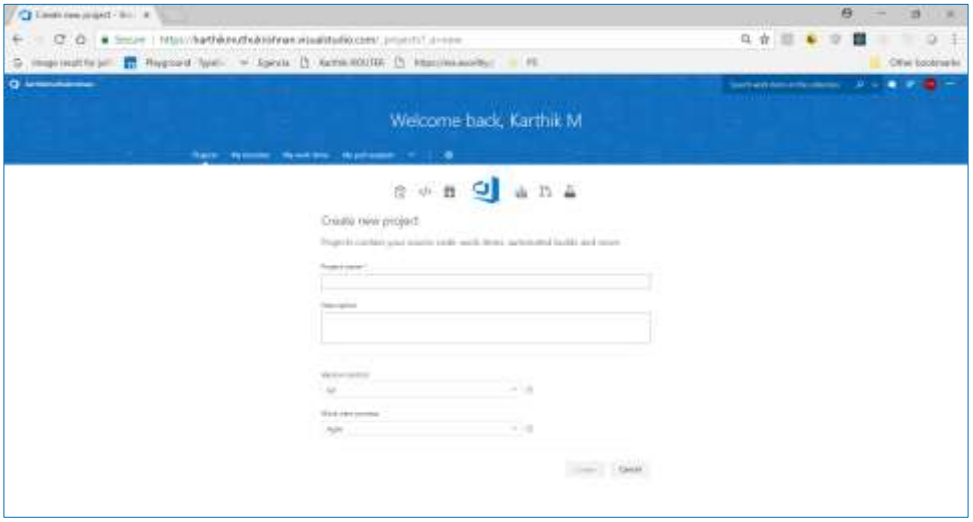
Sign up for free by creating an account and then, either upload your code to share or source control, or begin tracking your work using Scrum, Kanban, or a combination of methods.

Sign up	Add code to your repository
▪ Sign up with Team Services ▪ Create a team project ▪ Invite others to join your team	▪ Work in your favorite IDE: Xcode, Eclipse, Android Studio, IntelliJ, Visual Studio, Visual Studio Code

Getting started with VSTS – Creating Account



Creating a Team Services project



VSTS vs TFS

VSTS	TFS
VSTS is the cloud offering that provides a scalable, reliable, and globally available hosted service.	Team Foundation Server is the on-premises offering built on a SQL Server backend.
With VSTS, you connect over the public internet	With TFS you typically connect to an intranet server
VSTS, you can use a similar mechanism to provide access to groups of users by adding Azure AD groups to TFS groups.	TFS, you provide users access to deployments by adding Active Directory (AD) groups to various TFS groups

VSTS and Team Foundation Server (TFS) both provides an integrated, collaborative environment that supports Git, continuous integration, and Agile tools for planning and tracking work.

Summary

- VSTS basically TFS in the cloud
- Team Foundation Version Control(TFVC) and Git are the two types of version control systems available in VSTS
- TFVC is centralized version control where as Git is a distributed version control
- VSTS provides an integrated set of features that you access through your web browser or IDE client
- Build and Release are two of the DevOps services in Team Services that help you manage continuous integration and continuous delivery

People matter, results count.

About Capgemini

With more than 190,000 people, Capgemini is present in over 40 countries and celebrates its 50th Anniversary year in 2017. A global leader in consulting, technology and outsourcing services, the Group reported 2016 global revenues of EUR 12.5 billion. Together with its clients, Capgemini creates and delivers business, technology and digital solutions that fit their needs, enabling them to achieve innovation and competitiveness. A deeply multicultural organization, Capgemini has developed its own way of working, the Collaborative Business Experience™, and draws on Rightshore®, its worldwide delivery model.

Learn more about us at

www.capgemini.com