

Syllabus

Course Overview

In this course, you will learn and practice DevOps using Visual Studio Team Services, Visual Studio IDE, and the Azure portal. You will learn more about the principles of Continuous Integration (CI) and Continuous Delivery (CD), examining and applying many of the practices of DevOps such as source code management, build artifact engineering, release management, and deployment. You will apply these practices individually and collectively as a DevOps pipeline from the developer IDE all the way up to the Azure cloud deployment.

Course Competencies

(Read Only)

To successfully complete this course, you will be expected to:

- 1 Explain how DevOps principles and tools are applied in a cloud development and delivery platform.
- 2 Implement key practices of Continuous Integration (CI) of software.
- 3 Implement key practices of Continuous Delivery (CD) of software.
- 4 Apply tools and parameters to DevOps practices.
- 5 Automate software deployment using a continuous deployment pipeline.
- 6 Communicate effectively.

Course Prerequisites

IT4751

The materials listed below are required to complete the learning activities in this course.

Integrated Materials

Many of your required books are available via the VitalSource Bookshelf link in the courseroom, located in your Course Tools. Registered learners in a Resource Kit program can access these materials using the courseroom link on the Friday before the course start date. Some materials are available only in hard-copy format or by using an access code. For these materials, you will receive an email with further instructions for access. Visit the [Course Materials](#) page on Campus for more information.

Hardware

Capella University requires learners to meet certain minimum [computer requirements](#). The following hardware may go beyond those minimums and is required to complete learning activities in this course.

Note: If you already have the following hardware, you do not need to purchase it. Visit the [Course Materials](#) page on Campus for more information.

Hardware

External or built-in microphone

Library

The following required readings are provided in the Capella University Library or linked directly in this course. To find specific readings by journal or book title, use [Journal and Book Locator](#). Refer to the [Journal and Book Locator library guide](#) to learn how to use this tool.

- Olausson, M., & Ehn, J. (2015). [Continuous Delivery with Visual Studio ALM 2015 \(1st ed.\)](#). Apress.

External Resource

Please note that URLs change frequently. While the URLs were current when this course was designed, some may no longer be valid. If you cannot access a specific link, contact your instructor for an alternative URL. Permissions for the following links have been either granted or deemed appropriate for educational use at the time of course publication.

- Git (n.d.). [Git downloads](#). Retrieved from <https://git-scm.com/downloads>
- Github.com (n.d.). [Java-sample](#). Retrieved from <https://github.com/Adventworks/java-sample>
- Microsoft Azure. (2017). [Get started guide for Azure developers](#). Retrieved from <https://docs.microsoft.com/en-us/azure/guides/developer/azure-developer-guide>
- Microsoft Developer. (2018). [Taking control over your releases with VSTS Release Management \(RM\) Release Gates](#). Retrieved from https://www.youtube.com/watch?v=7WLCqwhTZ_4
- Microsoft [Visual Studio Team Services – Release Management. \[Video\]](#). Retrieved from <https://www.youtube.com/watch?v=zSPuRXTeZW8>
- Microsoft Visual Studio. (2016). [Azure Application Platform Overview](#). Retrieved from <https://www.youtube.com/watch?v=HQM4OJ9N0rg>

- Microsoft Visual Studio. (2016). [Release Management – Azure Deployments](https://www.youtube.com/watch?v=fd3WLghC1y0). Retrieved from <https://www.youtube.com/watch?v=fd3WLghC1y0>
- Microsoft Visual Studio. (2017). [Build a CI/CD pipeline from Visual Studio Team Services](https://www.youtube.com/watch?v=yNESSInONyk). Retrieved from <https://www.youtube.com/watch?v=yNESSInONyk>
- Microsoft Visual Studio. (2017). [Build a CI/CD pipeline from Visual Studio](https://www.youtube.com/watch?v=JhSG7Jbds7I). Retrieved from <https://www.youtube.com/watch?v=JhSG7Jbds7I>
- Microsoft Visual Studio. (2017). [Get started with Git in Visual Studio Team Services | T178](https://youtu.be/dVoAWTNojgg). Retrieved from <https://youtu.be/dVoAWTNojgg>
- Microsoft Visual Studio. (2017). [Using pull requests with Visual Studio Team Services | T184](https://youtu.be/tFB_O9FCh08). Retrieved from https://youtu.be/tFB_O9FCh08
- Microsoft Visual Studio. (2018). [Build and Release - 5 Minute Quickstarts](https://docs.microsoft.com/en-us/vsts/build-release/test/example-continuous-testing?view=vsts). Retrieved from <https://docs.microsoft.com/en-us/vsts/build-release/test/example-continuous-testing?view=vsts>
- Microsoft. (2017). [Build your ASP.NET 4 app](https://docs.microsoft.com/en-us/vsts/pipelines/apps/aspnet/build-aspnet-4?view=vsts&tabs=vsts). Retrieved from <https://docs.microsoft.com/en-us/vsts/pipelines/apps/aspnet/build-aspnet-4?view=vsts&tabs=vsts>
- Microsoft. (2017). [Install and set up Git](https://docs.microsoft.com/en-us/azure/devops/git/install-and-set-up-git). Retrieved from <https://docs.microsoft.com/en-us/azure/devops/git/install-and-set-up-git>
- Microsoft. (2017). [Learn Git with Team Services](https://docs.microsoft.com/en-us/azure/devops/git/learn-git-with-team-services). Retrieved from <https://docs.microsoft.com/en-us/azure/devops/git/learn-git-with-team-services>
- Microsoft. (2017). [Quickstart: Projects and solutions](https://docs.microsoft.com/en-us/visualstudio/ide/quickstart-projects-solutions). Retrieved from <https://docs.microsoft.com/en-us/visualstudio/ide/quickstart-projects-solutions>
- Microsoft. (2017). [Quickstart: Use the code editor](https://docs.microsoft.com/en-us/visualstudio/ide/quickstart-editor). Retrieved from <https://docs.microsoft.com/en-us/visualstudio/ide/quickstart-editor>
- Microsoft. (2017). [Save and share code with Git](https://docs.microsoft.com/en-us/azure/devops/git/git-share-code). Retrieved from <https://docs.microsoft.com/en-us/azure/devops/git/git-share-code>
- Microsoft. (2017). [Set up a repository](https://docs.microsoft.com/en-us/azure/devops/git/set-up-a-git-repository). Retrieved from <https://docs.microsoft.com/en-us/azure/devops/git/set-up-a-git-repository>
- Microsoft. (2017). [Understand Git history](https://docs.microsoft.com/en-us/azure/devops/git/understand-git-history). Retrieved from <https://docs.microsoft.com/en-us/azure/devops/git/understand-git-history>
- Microsoft. (2017). [Use Visual Studio to automatically generate a CI/CD pipeline to deploy your ASP.NET web app to Azure](https://docs.microsoft.com/en-us/vsts/build-release/archive/apps/aspnet/aspnet-4-ci-cd-vs-automatic?view=vsts). Retrieved from <https://docs.microsoft.com/en-us/vsts/build-release/archive/apps/aspnet/aspnet-4-ci-cd-vs-automatic?view=vsts>
- Microsoft. (2017). [What is Continuous Delivery?](https://www.visualstudio.com/learn/what-is-continuous-delivery/) Retrieved from <https://www.visualstudio.com/learn/what-is-continuous-delivery/>
- Microsoft. (2017). [What is Continuous Integration?](https://www.visualstudio.com/learn/what-is-continuous-integration/) Retrieved from <https://www.visualstudio.com/learn/what-is-continuous-integration/>
- Microsoft. (2017). [What is DevOps Culture?](https://docs.microsoft.com/en-us/azure/devops/what-is-devops-culture) Retrieved from <https://docs.microsoft.com/en-us/azure/devops/what-is-devops-culture>
- Microsoft. (2017). [What is version Control?](https://docs.microsoft.com/en-us/azure/devops/git/what-is-version-control) Retrieved from <https://docs.microsoft.com/en-us/azure/devops/git/what-is-version-control>
- Microsoft. (2018). [.NET Core](https://docs.microsoft.com/en-us/vsts/build-release/apps/aspnet/build-aspnet-core?view=vsts&tabs=gitvsts%2Cweb%2Cdeploy-windows). Retrieved from <https://docs.microsoft.com/en-us/vsts/build-release/apps/aspnet/build-aspnet-core?view=vsts&tabs=gitvsts%2Cweb%2Cdeploy-windows>
- Microsoft. (2018). [Build multiple branches](https://docs.microsoft.com/en-us/vsts/build-release/actions/ci-build-git?view=vsts). Retrieved from <https://docs.microsoft.com/en-us/vsts/build-release/actions/ci-build-git?view=vsts>

- Microsoft. (2018). [Build your Java app with Maven](https://docs.microsoft.com/en-us/vsts/pipelines/apps/java/build-maven?view=vsts&tabs=vsts). Retrieved from https://docs.microsoft.com/en-us/vsts/pipelines/apps/java/build-maven?view=vsts&tabs=vsts
- Microsoft. (2018). [Create a new Git repo in your project](https://docs.microsoft.com/en-us/vsts/git/create-new-repo?view=vsts). Retrieved from https://docs.microsoft.com/en-us/vsts/git/create-new-repo?view=vsts
- Microsoft. (2018). [Create a project](https://docs.microsoft.com/en-us/vsts/organizations/projects/create-project?view=vsts&tabs=new-nav). Retrieved from https://docs.microsoft.com/en-us/vsts/organizations/projects/create-project?view=vsts&tabs=new-nav
- Microsoft. (2018). [Create your first build and release](https://docs.microsoft.com/en-us/vsts/pipelines/get-started-designer). Retrieved from https://docs.microsoft.com/en-us/vsts/pipelines/get-started-designer
- Microsoft. (2018). [Get started with Git and VSTS](https://docs.microsoft.com/en-us/vsts/git/gitquickstart?view=vsts&tabs=visual-studio). Retrieved from https://docs.microsoft.com/en-us/vsts/git/gitquickstart?view=vsts&tabs=visual-studio
- Microsoft. (2018). [Improve code quality with branch policies](https://docs.microsoft.com/en-us/vsts/git/branch-policies?view=vsts). Retrieved from https://docs.microsoft.com/en-us/vsts/git/branch-policies?view=vsts
- Microsoft. (2018). [Install Visual Studio 2017](https://docs.microsoft.com/en-us/visualstudio/install/install-visual-studio). Retrieved from https://docs.microsoft.com/en-us/visualstudio/install/install-visual-studio
- Microsoft. (2018). [Quickstart: first look at the Visual Studio IDE](https://docs.microsoft.com/en-us/visualstudio/ide/quickstart-ide-orientation). Retrieved from https://docs.microsoft.com/en-us/visualstudio/ide/quickstart-ide-orientation
- Microsoft. (2018). [Release approvals and gates overview](https://docs.microsoft.com/en-us/vsts/pipelines/release/approvals/?view=vsts). Retrieved from https://docs.microsoft.com/en-us/vsts/pipelines/release/approvals/?view=vsts
- Microsoft. (2018). [Release deployment control using gates](https://docs.microsoft.com/en-us/vsts/pipelines/release/approvals/gates?view=vsts). Retrieved from https://docs.microsoft.com/en-us/vsts/pipelines/release/approvals/gates?view=vsts
- Microsoft. (2018). [Review code with pull requests](https://docs.microsoft.com/en-us/vsts/git/pull-requests?view=vsts). Retrieved from https://docs.microsoft.com/en-us/vsts/git/pull-requests?view=vsts
- Microsoft. (2018). [Share your code with Visual Studio 2017 and VSTS Git](https://docs.microsoft.com/en-us/vsts/git/share-your-code-in-git-vs-2017?view=vsts). Retrieved from https://docs.microsoft.com/en-us/vsts/git/share-your-code-in-git-vs-2017?view=vsts
- Microsoft. (2018). [TESTRelease deployment control using approvals](https://docs.microsoft.com/en-us/vsts/pipelines/release/approvals/approvals?view=vsts). Retrieved from https://docs.microsoft.com/en-us/vsts/pipelines/release/approvals/approvals?view=vsts
- Microsoft. (2018). [VSTS Git tutorial](https://docs.microsoft.com/en-us/vsts/git/tutorial/gitworkflow?view=vsts). Retrieved from https://docs.microsoft.com/en-us/vsts/git/tutorial/gitworkflow?view=vsts
- Microsoft. (2018). [What is DevOps?](https://docs.microsoft.com/en-us/azure/devops/what-is-devops) Retrieved from https://docs.microsoft.com/en-us/azure/devops/what-is-devops
- Microsoft. (2018). [What is Git?](https://docs.microsoft.com/en-us/azure/devops/git/what-is-git) Retrieved from https://docs.microsoft.com/en-us/azure/devops/git/what-is-git
- Microsoft. (n.d.). [Build and release - Videos](https://docs.microsoft.com/en-us/vsts/pipelines/?view=vsts#videos). Retrieved from https://docs.microsoft.com/en-us/vsts/pipelines/?view=vsts#videos
- Microsoft. (n.d.). [Sign up for a free VSTS organization and invite others to join your team](https://docs.microsoft.com/en-us/vsts/user-guide/sign-up-invite-teammates?view=vsts). Retrieved from https://docs.microsoft.com/en-us/vsts/user-guide/sign-up-invite-teammates?view=vsts
- Microsoft. (n.d.). [Visual Studio Team Services](https://www.visualstudio.com/team-services/). Retrieved from https://www.visualstudio.com/team-services/

Suggested

The following materials are recommended to provide you with a better understanding of the topics in this course. These materials are not required to complete the course, but they are aligned to course activities and assessments and are highly recommended for your use.

Optional

The following optional materials are offered to provide you with a better understanding of the topics in this course. These materials are not required to complete the course.

Unit 1 >> Introduction to DevOps

Introduction

Unit 1 introduces DevOps, its characteristics, and its benefits to organizations. You will learn about the Continuous Integration (CI) and Continuous Delivery (CD) of DevOps. There is no assignment for Unit 1.

Learning Activities

u01s1 - Software Preparation and Technology Access

In this course, you will be using software and technology that is needed to complete designated activities and assignments. There is no additional cost for this software and technology. Some software packages will be made available to you at no additional cost through Capella's subscription with Microsoft, while other software packages are available for free download through open-source licensing.

Capella University requires learners to meet certain minimum [computer requirements](#). Please note that some software required for a course may exceed these minimum requirements. Check the requirements for the software you may need to download and install to make sure it will work on your device. Most software will require a Windows PC. If you use a Mac, refer to [Installing a Virtual Environment and Windows on a Mac](#).

If you use assistive technology or any alternative communication methods to access course content, please contact DisabilityServices@Capella.edu with any access-related questions or to request accommodations.

For this course, follow the instructions provided through the links below to download and install software or register for an account, as required.

Note: As a Capella learner, you have access to IT online resources through Capella's Skillssoft subscription.

Azure Cloud

The hands-on labs in this course use the Microsoft Azure platform. Follow this link to set up your MS Azure account: [Setting up a Microsoft Azure Student Account](#).

Note: You likely already have an Azure account if you have taken IT4751. If so, continue to use that account.

MS Visual Studio

After you set up your Azure Cloud account, install MS Visual Studio Enterprise using the product key you found in the [Capella Microsoft Imagine WebStore](#).

Visual Studio Team Services (VSTS)

You will use Visual Studio Team Services to complete activities in this course. Sign up for your [free VSTS account here](#).

u01s2 - Studies

Readings

Study these resources to learn the CI/CD tools in the Visual Studio Team Services (VSTS).

Read the following Skillsoft resource:

- Olausson, M., & Ehn, J. (2015). [Continuous Delivery with Visual Studio ALM 2015](#) (1st ed.). Apress.
 - Chapter 1: "Introduction to Continuous Delivery," pp 1–14.

Read the following Internet resources:

- Microsoft. (2018). [What is DevOps?](#) Retrieved from <https://docs.microsoft.com/en-us/azure/devops/what-is-devops>
- Microsoft. (2018). [What is Continuous Integration?](#) Retrieved from <https://www.visualstudio.com/learn/what-is-continuous-integration/>
- Microsoft. (2018). [What is Continuous Delivery?](#) Retrieved from <https://www.visualstudio.com/learn/what-is-continuous-delivery/>
- Microsoft. (2018). [What is DevOps Culture?](#) Retrieved from <https://www.visualstudio.com/learn/what-is-continuous-delivery/>

u01s2 - Learning Components

- Create a VSTS project with Git repo.
- Edit file in code hub.
- Initialize a Git repo.
- Navigate to VSTS code hub.

- Save and queue a new build.
- Create a new build definition.
- Select build tasks.
- Create environments for the release.
- Select release tasks and enable continuous delivery.
- Navigate to release logs.
- Navigate within release logs.
- Save edited code in VSTS code hub.

u01s3 - Kaltura Media Preparation (optional)

The Unit 5 assignment requires you to record audio and capture your screen for a presentation. You **may choose** to use Kaltura Media software. Refer to the [Using Kaltura \[PDF\]](#) tutorial for directions on recording and submitting your recording in the courseroom.

If you have not already done so, set up and test your headset, using the installation instructions provided by the manufacturer. Then practice using it to ensure the audio quality is sufficient.

Note: If you require the use of assistive technology or alternative communication methods to participate in this activity, please contact DisabilityServices@Capella.edu to request accommodations.

u01d1 - DevOps Characteristics, Benefits, and Tasks

DevOps is the modern way of developing and deploying software. Understanding DevOps and how it works is the first step in this new and exciting pattern of developing and deploying software.

Discuss the following:

- The characteristics of DevOps that distinguish it from traditional configuration management.
- The benefits of DevOps.
- Two of the common tasks that are executed in a typical DevOps scenario.

As with all discussions in this course, it is recommended that you make your initial post by 12 a.m. Central Standard Time on Wednesday evening to allow time for your peers to respond.

Response Guidelines

The expectation within the course discussions is to respond to at least two posts by each Sunday evening, but it is highly recommended that you extend the dialog further. Responding over multiple days will help in stimulating a lively discussion. Start by submitting your initial post on (or before) Wednesday. If you provide responses to your peers on Thursday, Saturday, and Sunday, and your peers reciprocate with their responses, you will have more opportunity for in-depth interaction with your classmates and the instructor.

Prior to Sunday evening, read your peers' posts, and provide feedback to at least two of them.

Course Resources

Undergraduate Discussion Participation Scoring Guide

u01d1 - Learning Components

- Create a new build definition.
- Understand basic Git repository commands.
- Understand code reviews and branch policies.
- Navigate to VSTS Build & Release menu.
- Explain how pull request improve operational efficiency.

Unit 2 >> DevOps Continuous Integration/Continuous Delivery (CI/CD) with VSTS

Introduction

Unit 2 examines how DevOps services are provided in the Microsoft Visual Studio Team Services. You will develop your first very simple DevOps pipeline using VSTS in this unit.

Learning Activities

u02s1 - Studies

Reading

Read the following on the Internet and Skillsoft resources:

Learn the CI/CD tools in the Visual Studio Team Services (VSTS):

- Microsoft. (2018). [Create a project](https://docs.microsoft.com/en-us/vsts/organizations/projects/create-project?view=vsts&tabs=new-nav). Retrieved from <https://docs.microsoft.com/en-us/vsts/organizations/projects/create-project?view=vsts&tabs=new-nav>
- Microsoft. (2017). [Build your ASP.NET 4 app](https://docs.microsoft.com/en-us/vsts/pipelines/apps/aspnet/build-aspnet-4?view=vsts&tabs=vsts). Retrieved from <https://docs.microsoft.com/en-us/vsts/pipelines/apps/aspnet/build-aspnet-4?view=vsts&tabs=vsts>

- Olausson, M., & Ehn, J. (2015). [Continuous Delivery with Visual Studio ALM 2015](#) (1st ed.). Apress.
 - Chapter 2 – Overview of Visual Studio 2015 ALM, pp. 15–32.

Learn how to use the tools of VSTS:

- Microsoft. (2017). [Sign up for a free VSTS organization and invite others to join your team](#). Retrieved from <https://docs.microsoft.com/en-us/vsts/user-guide/sign-up-invite-teammates?view=vsts>
- Microsoft. (2018). [CI/CD Hello world](#). Retrieved from <https://docs.microsoft.com/en-us/vsts/build-release/actions/ci-cd-part-1?toc=/vsts/user-guide/toc.json&bc=/vsts/user-guide/breadcrumb/toc.json&view=vsts>
 - View this tutorial to learn how to setup a simple DevOps CI/CD pipeline.

u02s1 - Learning Components

- Create a VS project.
- Use VS Git repos and commands.
- Describe basic VSTS procedures.

u02d1 - Setup Your Free VSTS Account & Add Users

The Visual Studio Team Services is Microsoft's tool for DevOps CI/CD implementation. You will use VSTS throughout this course to manage source code, create build artifacts, conduct code reviews, manage releases and deploy to Azure. Learning how to apply VSTS tools is essential to successful DevOps practices.

In this discussion, you will set up your free account (subscription) to VSTS and invite others to join your VSTS team so that they can be your reviewers when conducting code reviews.

Complete these steps:

1. Login into VSTS using a Microsoft account.
2. Create a VSTS account (subscription). You will use this account to create several projects during this course. You may call this account something like IT4753.
3. Create a new VSTS project in your newly created account.
4. Add at least one other member (besides you) to your VSTS list of users. You will later ask the added users to review your code via a Git pull request.

Provide two screenshots of your work result in your post. The first screenshot should show successful creation of your VSTS account. The second screenshot should show successful adding of at least one member to your VSTS list of users. See the Discussion u02d1 Screenshots file in the Resources for examples.

Discuss how you completed this exercise, the major issues you face and how you solved them, reflect on your experience and what you learned.

As with all discussions in this course, it is recommended that you make your initial post by 12 a.m. Central Standard Time on Wednesday evening to allow time for your peers to respond.

Response Guidelines

The expectation within the course discussions is to respond to at least two posts by each Sunday evening, but it is highly recommended that you extend the dialog further. Responding over multiple days will help in stimulating a lively discussion. Start by submitting your initial post on (or before) Wednesday. If you provide responses to your peers on Thursday, Saturday, and Sunday, and your peers reciprocate with their responses, you will have more opportunity for in-depth interaction with your classmates and the instructor.

Prior to Sunday evening, read your peers' posts, and provide feedback to at least two of them.

Course Resources

[Undergraduate Discussion Participation Scoring Guide](#)

[Discussion u02d1 Screenshots](#)

u02d1 - Learning Components

- [Navigate to VSTS Build & Release menu.](#)

u02a1 - DevOps Continuous Integration/Continuous Delivery (CI/CD) with VSTS

Overview

Visual Studio Team Services (VSTS) is a cloud-based Software as a Service (SaS) application used to streamline DevOps. VSTS provides many online tools to manage source code, engineer builds, and manage releases. Additionally, VSTS provides features to automate these DevOps practices with little manual effort. Applying these VSTS tools will help you accomplish your DevOps tasks quickly, efficiently, and effectively.

In this assignment you will set up a simple automated VSTS DevOps pipeline exercising its source code management, build, and release tools. You will create a simple PowerShell script (that prints out your name) and use it as the source code for your DevOps pipeline. After modifying the PowerShell script, your DevOps pipeline will automatically deploy this PowerShell script to two different deployment environments. You will also explain how you used the various VSTS tools to implement and validate the pipeline.

Preparation

- [Sign up for a free VSTS account from the link found in the Resources.](#)

- Download the Assignment Submission Template from the Resources and use it to create and submit your assignment.

Directions

Follow the directions in the Unit 2 Assignment Directions document found in the Resources.

Submission Requirements

Complete and submit all work specified in the directions using the Assignment Submission Template. Make sure to rename it using the appropriate unit number.

Course Resources

Assignment Submission Template

Unit 2 Assignment Directions

[Sign Up for a Free VSTS Organization and Invite Others to Join Your Team](#)

Unit 3 >> Managing Source Code with VSTS & Local Command Line Git

Introduction

Unit 3 introduces the very important practice of source code management. You will learn about the distributed source code management system called Git and its commands. You will manage source code using Git in both VSTS and your local Git command line.

Learning Activities

u03s1 - Studies

Reading

Read the following on the Internet:

- Microsoft. (2017). [Install and set up Git](https://docs.microsoft.com/en-us/azure/devops/git/install-and-set-up-git). Retrieved from https://docs.microsoft.com/en-us/azure/devops/git/install-and-set-up-git

- Microsoft. (2018). [VSTS Git tutorial](https://docs.microsoft.com/en-us/vsts/git/tutorial/gitworkflow?view=vsts). Retrieved from https://docs.microsoft.com/en-us/vsts/git/tutorial/gitworkflow?view=vsts
- Microsoft. (2017). [Learn Git with Team Services](https://docs.microsoft.com/en-us/azure/devops/git/learn-git-with-team-services). Retrieved from https://docs.microsoft.com/en-us/azure/devops/git/learn-git-with-team-services
- Microsoft. (2018). [Create a new Git repo in your project](https://docs.microsoft.com/en-us/vsts/git/create-new-repo?view=vsts). Retrieved from https://docs.microsoft.com/en-us/vsts/git/create-new-repo?view=vsts
- Microsoft. (2017). [What is version control?](https://docs.microsoft.com/en-us/azure/devops/git/what-is-version-control) Retrieved from https://docs.microsoft.com/en-us/azure/devops/git/what-is-version-control
- Microsoft. (2018). [What is Git?](https://docs.microsoft.com/en-us/azure/devops/git/what-is-git) Retrieved from https://docs.microsoft.com/en-us/azure/devops/git/what-is-git
- Microsoft. (2017). [Set up a repository](https://docs.microsoft.com/en-us/azure/devops/git/set-up-a-git-repository). Retrieved from https://docs.microsoft.com/en-us/azure/devops/git/set-up-a-git-repository
- Microsoft. (2017). [Save and share code with Git](https://docs.microsoft.com/en-us/azure/devops/git/git-share-code). Retrieved from https://docs.microsoft.com/en-us/azure/devops/git/git-share-code
- Microsoft. (2017). [Understand Git history](https://docs.microsoft.com/en-us/azure/devops/git/understand-git-history). Retrieved from https://docs.microsoft.com/en-us/azure/devops/git/understand-git-history

Read the following Skillsoft resource:

- Olausson, M., & Ehn, J. (2015). [Continuous Delivery with Visual Studio ALM 2015](#) (1st ed.). Apress.
 - Chapter 5: "Source Control Management," pp. 65–86.

u03d1 - Manage HTML Code Using VSTS Git

Git has become the de facto standard code management and version control system. It is widely available and used in many software development platforms including VSTS. Learning how to use Git is a prerequisite to all DevOps practices.

In this discussion, you will use VSTS Git to manage HTML files in a Git repo using a typical Git workflow.

Sign in to the free VSTS account you created in the previous unit.

Complete these steps:

1. Commit the content of a new HTML file to new Git repo master branch using VSTS. The content of the HTML file is a greeting like this: `<html><body>Hello World!</body></html>`
2. Commit changes to the HTML file to a branch of the master branch using VSTS. The changes are to replace the greeting with a different greeting. For example, from [Hello World!] to [Hello *YourName*].
3. Merge the changed branch into the master branch using VSTS pull request.

Provide three screenshots of your result of your work. The first screenshot should show the content of the HTML file in the master branch. The second screenshot should show the content of the HTML file in the updated branch.

The third screenshot should show the history of commits in your VSTS project. See the Discussion u03d1 Screenshots file in the Resources for examples.

Briefly discuss how you completed this exercise, the major issues you face and how you solved them, reflect on your experience and what you learned.

As with all discussions in this course, it is recommended that you make your initial post by 12 a.m. Central Standard Time on Wednesday evening to allow time for your peers to respond.

Response Guidelines

The expectation within the course discussions is to respond to at least two posts by each Sunday evening, but it is highly recommended that you extend the dialog further. Responding over multiple days will help in stimulating a lively discussion. Start by submitting your initial post on (or before) Wednesday. If you provide responses to your peers on Thursday, Saturday, and Sunday, and your peers reciprocate with their responses, you will have more opportunity for in-depth interaction with your classmates and the instructor.

Prior to Sunday evening, read your peers' posts, and provide feedback to at least two of them.

Course Resources

Undergraduate Discussion Participation Scoring Guide

[Sign Up for a Free VSTS Organization and Invite Others to Join Your Team](#)

Discussion u03d1 Screenshots

u03d1 - Learning Components

- Create a VSTS project with Git repo.
- Use Git clone command and its parameters.
- Use Git command line commands.

u03a1 - Managing Source Code with VSTS & Local Command Line Git

Overview

A Git repository (repo) is typically the starting point for any DevOps practices. Git is the most popular and widely used source code management system. Git is a distributed system allowing Git repos to share and synchronize code. Git has a standard workflow consisting of a number of commands (e.g. init, clone, commit, and push) that make it easy to share and synchronize code. Following the Git workflow and applying its commands will help you start your VSTS CI/CD pipeline on the right path.

In this assignment, you manage source code using VSTS and local Git command line exercising several Git commands like clone, commit, and push. You create a Git repo in VSTS. Using local Git command line workflow commands, you update a file in the VSTS repo. You also explain how you used the VSTS and Git command line tools to implement your Git workflow.

Preparation

- Download and install Git local on your computer (Windows, Mac, or Linux) from the Git Downloads link in the Resources.
- Download the Assignment Submission Template from the Resources and use it to create and submit your assignment.

Directions

Follow the directions in the Unit 3 Assignment Directions document found in the Resources.

Submission Requirements

Complete and submit all work specified in the directions using the Assignment Submission Template. Make sure to rename it using the appropriate unit number.

Course Resources

[Git downloads.](#)

Unit 3 Assignment Directions

Assignment Submission Template

Unit 4 >> Managing Source Code of Git Branches Using VSTS and VS

Introduction

Unit 4 extends the learning about Git into VS. You will learn how to use Git in VS and VSTS. You will also learn how to manage source code in branches using Git. You will manage the source code of an ASP.NET Core application using Git in VS and VSTS.

Learning Activities

Reading

Read the following on the Internet:

- Microsoft. (2018). [Install Visual Studio 2017](https://docs.microsoft.com/en-us/visualstudio/install/install-visual-studio). Retrieved from <https://docs.microsoft.com/en-us/visualstudio/install/install-visual-studio>
 - Study this resource to learn how install Visual Studio 2017 and its features.
- Microsoft. (2018). [Quickstart: First look at the Visual Studio IDE](https://docs.microsoft.com/en-us/visualstudio/ide/quickstart-ide-orientation). Retrieved from <https://docs.microsoft.com/en-us/visualstudio/ide/quickstart-ide-orientation>
- Microsoft. (2017). [Quickstart: Projects and solutions](https://docs.microsoft.com/en-us/visualstudio/ide/quickstart-projects-solutions). Retrieved from <https://docs.microsoft.com/en-us/visualstudio/ide/quickstart-projects-solutions>
 - How to install Visual Studio 2017 and its features.
- Microsoft. (2017). [Quickstart: Coding in the editor](https://docs.microsoft.com/en-us/visualstudio/ide/quickstart-editor). Retrieved from <https://docs.microsoft.com/en-us/visualstudio/ide/quickstart-editor>
 - A quickstart guide to coding in VS.
- Microsoft. (2018). [Share your code with Visual Studio 2017 and VSTS Git](https://docs.microsoft.com/en-us/vsts/git/share-your-code-in-git-vs-2017?view=vsts). Retrieved from <https://docs.microsoft.com/en-us/vsts/git/share-your-code-in-git-vs-2017?view=vsts>
 - How to create local Git repo in VS and to publish it to VSTS.
- Microsoft. (2018). [Get Started with Git and VSTS](https://docs.microsoft.com/en-us/vsts/git/gitquickstart?view=vsts&tabs=visual-studio). Retrieved from <https://docs.microsoft.com/en-us/vsts/git/gitquickstart?view=vsts&tabs=visual-studio>
 - How to use Git locally in VS and how to use it in VSTS.

u04s1 - Learning Components

- Create a VSTS project with Git repo.
- Navigate VSTS menus and screens.
- Understand differences in Navigate vs. Team Explorer.

u04d1 - Manage C# .NET Core Console Application Code Using VS & VSTS

The VS Git system allows you to manage your code locally and then push your updates to VSTS Git. Managing your Git repo locally first on VS affords you the ease and productivity of using the VS IDE to edit your code and makes it easy to update your team's Git repo on VSTS.

In this discussion, you manage a local Git repo for a C# console application and push changes to a VSTS repo.

By now you should have downloaded and installed VS 2017 from your Capella Imagine account.

Complete these steps:

1. Publish a newly created local repo master branch of a C# .NET Core console application to VSTS using VS. Leave the content of the new application unchanged.
2. Push committed changes to the "Program.cs" file of the application to VSTS using VS. The changes are to replace the greeting with a different greeting. For example, from [Hello World!] to [Hello IT4753!]
3. View a summary of the last push on the master branch using VSTS

Provide three screenshots of your result of your work. The first screenshot should show the content of the "Program.cs" file in the master branch in VSTS. The second screenshot should show the content of the "Program.cs" file after the local commits are pushed to VSTS. The third screenshot should show a summary of the last push showing the original and updated greeting in the "Program.cs" file in your VSTTS project. See the Discussion u04d1 Screenshots file in the Resources for examples.

Briefly discuss how you completed this exercise, the major issues you face and how you solved them, reflect on your experience and what you learned.

As with all discussions in this course, it is recommended that you make your initial post by 12 a.m. Central Standard Time on Wednesday evening to allow time for your peers to respond.

Response Guidelines

The expectation within the course discussions is to respond to at least two posts by each Sunday evening, but it is highly recommended that you extend the dialog further. Responding over multiple days will help in stimulating a lively discussion. Start by submitting your initial post on (or before) Wednesday. If you provide responses to your peers on Thursday, Saturday, and Sunday, and your peers reciprocate with their responses, you will have more opportunity for in-depth interaction with your classmates and the instructor.

Prior to Sunday evening, read your peers' posts, and provide feedback to at least two of them.

Course Resources

Undergraduate Discussion Participation Scoring Guide

Discussion u04d1 Screenshots

u04d1 - Learning Components

- Understand basic Git repository commands.

u04a1 - Managing Source Code Branches with VSTS & VS

Overview

Visual Studio (VS) supports the Git workflow. From within VS, you can execute many of the standard Git commands like clone, commit, and push. These VS Git commands are essentials to interfacing your local VS code development repo with your team repo in Visual Studio Team Services (VSTS).

In this assignment, you clone, commit, and push local Git changes on an ASP.NET Core code branch to VSTS using VS. You import a Git repo into VSTS and update a file in the VSTS repo branch using a typical Git workflow in your local VS. You also explain how you used the VSTS and VS Git tools to implement your Git workflow.

Preparation

Make sure you have download and installed VS 2017 from your Capella Imagine account.

Directions

Follow the directions in the Unit 4 Assignment Directions document found in the Resources.

Submission Requirements

Complete and submit all work specified in the directions using the Assignment Submission Template. Make sure to rename it using the appropriate unit number.

Course Resources

Assignment Submission Template

Unit 4 Assignment Directions

Unit 5 >> Conducting Code Reviews with Git Pull Requests

Introduction

Unit 5 addresses a very important practice of DevOps: conducting code reviews using Git pull requests. You will develop two short videos explaining Git pull requests to two different audiences.

Learning Activities

u05s1 - Studies

Reading and Video

Use the Internet to view the following resources:

- Microsoft Visual Studio. (2017). [Get started with Git in Visual Studio Team Services | T178](https://youtu.be/dVoAWTNojgg) [Video] | [Transcript](#). Retrieved from <https://youtu.be/dVoAWTNojgg> (9:37).

Read the following information related to Git Pull requests and branch policies in VSTS.

- Microsoft. (2018). [Review code with pull requests](https://docs.microsoft.com/en-us/vsts/git/pull-requests?view=vsts). Retrieved from <https://docs.microsoft.com/en-us/vsts/git/pull-requests?view=vsts>
- Microsoft. (2018). [Improve code quality with branch policies](https://docs.microsoft.com/en-us/vsts/git/branch-policies?view=vsts). Retrieved from <https://docs.microsoft.com/en-us/vsts/git/branch-policies?view=vsts>
- Microsoft Visual Studio. (2017). [Using pull requests with Visual Studio Team Services | T184](https://youtu.be/tFB_O9FCh08) [Video] | [Transcript](#). Retrieved from https://youtu.be/tFB_O9FCh08 (15:13).

u05s1 - Learning Components

- Understand code reviews and branch policies.
- Explain how pull request improve operational efficiency.

u05s2 - Kaltura Media (optional)

In preparation for creating the audio recordings required for this unit's assignment, complete the following **only if you plan to use Kaltura for your presentation**:

- If you have not already done so, set up and test your audio recording device on your computer, using the installation instructions from the manufacturer.
- Practice using the audio equipment to ensure the audio quality is sufficient.
- Refer to the [Using Kaltura](#) tutorial for directions on recording and uploading your recordings in the courseroom.

Note: If you require the use of assistive technology or alternative communication methods to participate in this activity, please contact DisabilityServices@Capella.edu to request accommodations.

u05d1 - Create and Complete JavaServer Pages (JSP) Pull Requests Requiring Two Reviewers Using VSTS

Git pull requests merge changes committed to a branch to the master branch. In addition, pull requests are great tools to conduct code reviews and to enforce branch policies on the master branch. Using pull requests for code reviews and branch policies will increase the quality of the software you develop.

In this discussion, you will create and complete a simple pull request for a JavaServer Pages (JSP) application. You will enforce a master branch policy of requiring a minimum of two reviewers to complete the pull request using VSTS.

Complete these steps:

1. Import a JavaServer Pages (JSP) Git repo into a new repo using VSTS. The JSP application repo to import is titled Java-sample, linked in the Resources for this discussion.
2. Commit changes to the "src/main/webapp/index.jsp" to a branch of the master branch. The changes to the "index.jsp" file and the name of the branch are up to you.
3. Setup a branch policy on the master branch requiring at least two reviewers to complete a pull request.
4. Create a pull request and adding at least one additional team member as reviewers. You can be the second reviewer. One of the reviewers must add a comment about the changes to the "index.jsp" file.
5. Once the two reviewers approved the changes, complete the pull request.

Provide two screenshots of your result of your work. The first screenshot should show the overview of the pull request after it has been approved by the two reviewers with comments but before the pull request was completed. The second screenshot should show the overview of the pull request once it was completed. See the Discussion u05d1 Screenshots file in the Resources for examples.

Briefly discuss how you completed this exercise, the major issues you face and how you solved them, reflect on your experience and what you learned.

As with all discussions in this course, it is recommended that you make your initial post by 12 a.m. Central Standard Time on Wednesday evening to allow time for your peers to respond.

Response Guidelines

The expectation within the course discussions is to respond to at least two posts by each Sunday evening, but it is highly recommended that you extend the dialog further. Responding over multiple days will help in stimulating a lively discussion. Start by submitting your initial post on (or before) Wednesday. If you provide responses to your peers on Thursday, Saturday, and Sunday, and your peers reciprocate with their responses, you will have more opportunity for in-depth interaction with your classmates and the instructor.

Prior to Sunday evening, read your peers' posts, and provide feedback to at least two of them.

Course Resources

Undergraduate Discussion Participation Scoring Guide

u05d1 - Learning Components

- Understand basic Git repository commands.
- Understand code reviews and branch policies.
- Explain how pull request improve operational efficiency.

u05a1 - Video: Git Pull Requests

Overview

As an IT professional, the ability to communicate effectively in a variety of professional context is essential to a successful IT career. Effective communication skills include understanding your audience, defining the goal of your message to your audience, and conveying the message clearly and precisely.

In this assignment you create a video that explains aspects of Git pull requests to an audience of peers and company management.

Preparation

- Choose a presentation software to create your presentation.
- Consider these guidelines as you prepare to create your presentation:
 - It is suggested that you write an outline or script of what you are going to say before you begin recording in addition to having your design and supporting visuals ready. Although many software programs allow you to pause or edit, it is advisable to prepare before you start recording.
 - View your video prior to posting to ensure that the audio volume is appropriate.

Kaltura (optional)

For this assignment, you may choose to create your presentation using Kaltura. To learn how to use Kaltura, refer to the Using Kaltura tutorial found in the second study in this unit.

Note: If you require the use of assistive technology or alternative communication methods to participate in these activities, please contact DisabilityServices@Capella.edu to request accommodations.

Directions

Create a 4-6 minute self-narrated video that informs the following two audiences about Git repo pull requests:

- Company Managers interested in understand the value of pull requests to the business.
- IT Peers interested in learning details about Git pull request workflow.

The video should:

- Introduce you and the goal of your presentation.
- Explain Git pull requests – what they are, how they work, and why they are important.
- Explain how Git pull requests enable code reviews and enforce branch policies.
- Describe pull requests workflow for IT teammates.
- Explain the business benefits of pull requests.
- Include visual elements appropriate for illustrating your narrative.
- Include a brief summary.

Submission Requirements

You may submit your assignment in either one of two ways:

- Submit a Word document with a link to your video on a specified web server of your choosing.
- Zip your video file and submit it in the courseroom.

Course Resources

DisabilityServices@Capella.edu

External or built-in microphone

1. External or built-in microphone

Unit 6 >> Engineering CI Builds Using VSTS

Introduction

Unit 6 examines build engineering in VSTS. You will learn how to define a build and how to use the build definition to generate a build artifact. There is no assignment for Unit 6.

Learning Activities

u06s1 - Studies

Reading

Read the following Internet resources:

- Microsoft. (2018). [.NET Core](https://docs.microsoft.com/en-us/vsts/build-release/apps/aspnet/build-aspnet-core?view=vsts&tabs=gitvsts%2Cweb%2Cdeploy-windows). Retrieved from <https://docs.microsoft.com/en-us/vsts/build-release/apps/aspnet/build-aspnet-core?view=vsts&tabs=gitvsts%2Cweb%2Cdeploy-windows>
- Microsoft. (2018). [Build your Java app with Maven](https://docs.microsoft.com/en-us/vsts/pipelines/apps/java/build-maven?view=vsts&tabs=vsts). Retrieved from <https://docs.microsoft.com/en-us/vsts/pipelines/apps/java/build-maven?view=vsts&tabs=vsts>

Read the following Skillsoft resource:

- Olausson, M., & Ehn, J. (2015). [Continuous Delivery with Visual Studio ALM 2015](#) (1st ed.). Apress.
 - Chapter 7: "Build Automation," pp. 113–182.

u06s1 - Learning Components

- Navigate VSTS menus and screens.
- Save and queue build definitions.
- Enable CI in build definition.
- Specify branches to include in triggers.
- Commit code changes in VSTS code hub.
- Edit a code hub file.

u06d1 - Manually Engineer Builds for JavaServer Pages (JSP) Using VSTS

One of the main practices of DevOps is to engineer and create a build artifact from the source code. The created artifact is usually setup for automatic generation as part of a DevOps pipeline, but it can also be manually created. Defining a build task and then creating a build artifact are two core skills of any DevOps practice.

In this discussion, you manually engineer a build for a build for a JavaServer Pages (JSP) application that uses Maven for building its build artifact.

Complete these steps:

1. Import a JavaServer Pages (JSP) Git repo into a new repo using VSTS. Refer to the Java-sample link in the Resources to import the repo.
2. Define a new build for the imported repo. Use the Maven template, Hosted VS2017 agent, and Enable continuous integration to define your build.
3. Manually create a build artifact.

Provide two screenshots of your result of your work. The first screenshot should show the completion of the execution of the build tasks in the console. The second screenshot should show a summary of the build showing the name of the build definition, the Git branch used for the build, and the agent queue name used for the execution of the build. See the Discussion u06d1 Screenshots file in the Resources for examples.

Briefly discuss how you completed this exercise, the major issues you face and how you solved them, reflect on your experience and what you learned.

As with all discussions in this course, it is recommended that you make your initial post by 12 a.m. Central Standard Time on Wednesday evening to allow time for your peers to respond.

Response Guidelines

The expectation within the course discussions is to respond to at least two posts by each Sunday evening, but it is highly recommended that you extend the dialog further. Responding over multiple days will help in stimulating a lively discussion. Start by submitting your initial post on (or before) Wednesday. If you provide responses to your peers on Thursday, Saturday, and Sunday, and your peers reciprocate with their responses, you will have more opportunity for in-depth interaction with your classmates and the instructor.

Prior to Sunday evening, read your peers' posts, and provide feedback to at least two of them.

Course Resources

[Undergraduate Discussion Participation Scoring Guide](#)

[Discussion u06d1 Screenshots](#)

[Java-sample](#)

u06d1 - Learning Components

- Define CI build agent and tasks.
- Navigate VSTS menus and screens.
- Save and queue build definitions.
- Enable CI in build definition.
- Specify branches to include in triggers.
- Commit code changes in VSTS code hub.
- Edit a code hub file.

Unit 7 >> Engineering CI Builds for Git Branches Using VSTS

Introduction

Unit 7 continues the examination of VSTS build engineering. You will learn how to define builds for branches in your Git repo and engineer a CI build for feature branches of the Git repo of a JSP application.

Learning Activities

Reading

Read the following on the Internet:

- Microsoft. (2018). [Define a continuous integration \(CI\) build process for your Git repo](https://docs.microsoft.com/en-us/vsts/build-release/actions/ci-build-git?view=vsts). Retrieved from <https://docs.microsoft.com/en-us/vsts/build-release/actions/ci-build-git?view=vsts>
 - Information on how to define and trigger builds for feature branches in your Git repo.

u07s1 - Learning Components

- Define CI build agent and tasks.
- Navigate VSTS menus and screens.
- Save and queue build definitions.
- Enable CI in build definition.
- Specify branches to include in triggers.
- Commit code changes in VSTS code hub.
- Edit a code hub file.

u07d1 - Define VSTS Build Definitions, Builds, and the Relationship Between Them

DevOps Continuous Integration (CI) is implemented in VSTS via build definitions from which build artifacts are generated. Understanding the meaning of these VSTS entities and how they relate to each other is vital for successful DevOps using VSTS.

Do the following:

1. Define what is meant by a VSTS build definition, the components of a build definition, and how you can create a build definition.
2. Define what is meant by a VSTS build and how you can create a build artifact.
3. Explain the relationship between VSTS build definitions and VSTS builds and represent this relationship as UML class diagram.

As with all discussions in this course, it is recommended that you make your initial post by 12 a.m. Central Standard Time on Wednesday evening to allow time for your peers to respond.

Response Guidelines

The expectation within the course discussions is to respond to at least two posts by each Sunday evening, but it is highly recommended that you extend the dialog further. Responding over multiple days will help in stimulating a lively discussion. Start by submitting your initial post on (or before) Wednesday. If you provide responses to your

peers on Thursday, Saturday, and Sunday, and your peers reciprocate with their responses, you will have more opportunity for in-depth interaction with your classmates and the instructor.

Prior to Sunday evening, read your peers' posts, and provide feedback to at least two of them.

Course Resources

Undergraduate Discussion Participation Scoring Guide

u07d1 - Learning Components

- Define CI build agent and tasks.
- Create a VSTS project with Git repo.
- Navigate VSTS menus and screens.
- Edit file in code hub.
- Save and queue build definitions.
- Initialize a Git repo.
- Enable CI in build definition.
- Navigate to VSTS code hub.
- Specify branches to include in triggers.
- Create a new build definition.
- Commit code changes in VSTS code hub.
- Edit a code hub file.

u07a1 - Defining CI Build Agents & Tasks for Applications

Overview

VSTS allows us to extend the CI build trigger to include code changes made on Git branches in addition to code changes made on the master branch. Building the branch code helps you catch errors earlier to ensure quality branch code that can be safely merged into the master branch.

In this assignment, you engineer CI builds for a JavaServer Pages (JSP) application to include branch code changes. You import a JavaServer Pages (JSP) Git repo into VSTS. You update a file in a VSTS repo branch using the same CI build of the master branch and explain how you use the VSTS build tools to implement your workflow.

Directions

Follow the directions in the Unit 7 Assignment Directions document found in the Resources.

Submission Requirements

Complete and submit all work specified in the directions using the Assignment Submission Template. Make sure to rename it using the appropriate unit number.

Course Resources

Assignment Submission Template

Unit 7 Assignment Directions

Unit 8 >> Managing CD Releases for Azure Deployment Using VS Extensions

Introduction

Unit 8 examines DevOps release management and deployment to Azure. You will learn about the VS extension for automatically building a DevOps pipeline from within the VS IDE and use this VS extension to automatically build a DevOps pipeline for an ASP.NET Core web application.

Learning Activities

u08s1 - Studies

Reading and Video

Learn about continuous delivery in VSTS and its features.

- Olausson, M., & Ehn, J. (2015). [Continuous Delivery with Visual Studio ALM](#) (1st ed.). Apress.
 - Chapter 10: "Building a Deployment Pipeline," pp. 279–305.
- Microsoft Visual Studio. (2016). [Visual Studio Team Services – Release Management](#) [Video] | [Transcript](#). Retrieved from <https://www.youtube.com/watch?v=zSPuRXTeZW8> (2:07).

Learn about the Visual Studio extension to create a CI/CD pipeline all the way to an Azure app service.

- Microsoft Visual Studio. (2017). [Build a CI/CD pipeline from Visual Studio](#) [Video] | [Transcript](#). Retrieved from <https://www.youtube.com/watch?v=JhSG7Jbds7I> (14:31).
- Microsoft. (2017). [Use Visual Studio to automatically generate a CI/CD pipeline to deploy your ASP.NET web app to Azure](#). Retrieved from <https://docs.microsoft.com/en-us/vsts/build-release/archive/apps/aspnet/aspnet-4-ci-cd-vs-automatic?view=vsts>

Study these resources to learn about the Azure services available to software developers.

- Microsoft. (2017). [Get started guide for Azure developers](https://docs.microsoft.com/en-us/azure/guides/developer/azure-developer-guide). Retrieved from <https://docs.microsoft.com/en-us/azure/guides/developer/azure-developer-guide>
- Microsoft Visual Studio. (2016). [Azure Application Platform Overview \[Video\]](#) | [Transcript](#). Retrieved from <https://www.youtube.com/watch?v=HQM4OJ9N0rg> (6:59).

u08s1 - Learning Components

- Create a VS project.
- • Install VS extensions.
- Use VS Git repos and commands.
- Describe basic VSTS procedures.
- Explain VS solutions and projects
- Use Git commit changes of code.
- Use VS IDE text editors.

u08d1 - Setup Your Imagine Azure Account

The enabling technology behind DevOps continuous Delivery and Continuous Deployment is the automation of creating immediate and scalable computer resources on which to deploy software releases. This ability is provided by many Cloud Service Providers (CSP) such Amazon AWS and Microsoft Azure. It is critical that you acquire cloud knowledge and skills to be effective in your DevOps practices.

In this discussion, you use your account on the Capella Microsoft Imagine webstore to setup, activate, and access your Imagine Azure account. **Note:** you may already have an Azure account from IT4751.

Complete these steps:

1. Register on the on the Capella Microsoft Imagine webstore and acquire the free Microsoft Azure for Student product.
2. Follow the directions given to setup your free Imagine Azure account.
3. Login into your Imagine Azure account and view your bill by going to Azure profile (on the upper right-hand corner) > View my bill > Overview.

Provide a screenshot of your result of your work. The screenshot should show your Microsoft account that you used for your Azure account and should use the details of your Imagine (DreamSpark) subscription. See the Discussion u08d1 Screenshots file in the Resources for examples.

Briefly discuss how you completed this exercise, the major issues you face and how you solved them, reflect on your experience and what you learned.

As with all discussions in this course, it is recommended that you make your initial post by 12 a.m. Central Standard Time on Wednesday evening to allow time for your peers to respond.

Response Guidelines

The expectation within the course discussions is to respond to at least two posts by each Sunday evening, but it is highly recommended that you extend the dialog further. Responding over multiple days will help in stimulating a lively discussion. Start by submitting your initial post on (or before) Wednesday. If you provide responses to your peers on Thursday, Saturday, and Sunday, and your peers reciprocate with their responses, you will have more opportunity for in-depth interaction with your classmates and the instructor.

Prior to Sunday evening, read your peers' posts, and provide feedback to at least two of them.

Course Resources

[Undergraduate Discussion Participation Scoring Guide](#)

[Discussion u08d1 Screenshots](#)

u08d1 - Learning Components

- Create manual and automatic deployment of releases to Azure
- Navigate VSTS menus and screens.
- Push and verify changes from VS to VSTS
- Create manual and automatic VSTS builds.
- Create Azure web apps.
- Use VS IDE text editors.

u08a1 - Managing CD Releases of Applications with VS

Overview

The Continuous Delivery Tools for Visual Studio is an extension to Visual Studio that makes it possible to create a complete CI/CD pipeline for your Visual Studio project. This extension creates a VSTS build definition, a VSTS release definition, and an Azure app service to which to deploy the releases of your VS project. You can apply this VS extension to quickly setup and test your VS project on Azure without having to define build and release definition and without having to manually create an app service for your release deployment.

In this assignment, you create and trigger an automated CI/CD pipeline, that includes deployment to Azure, using the VS continuous delivery tools extension. You configure continuous delivery for an ASP.NET Core application in VS (After making code changes to the VS project, your CI/CD pipeline automatically deploy the changed

application to Azure). Lastly, you explain how you used the VS extension to implement and validate the CI/CD pipeline.

Directions

Follow the directions in the Unit 8 Assignment Directions document found in the Resources.

Submission Requirements

Complete and submit all work specified in the directions using the Assignment Submission Template. Make sure to rename it using the appropriate unit number.

Course Resources

Unit 8 Assignment Directions

Assignment Submission Template

Unit 9 >> Managing CD Releases for Azure Deployment Using VS, VSTS, and Azure Portal

Introduction

Unit 9 addresses a deeper aspect of release management in VSTS and how to configure the DevOps pipeline manually. You will manage releases of an ASP.NET Core application using VS, VSTS, and the Azure portal.

Learning Activities

u09s1 - Studies

Reading

Read the following on the Internet:

- Microsoft Visual Studio. (2018). [Build and release - 5 Minute Quickstarts](https://docs.microsoft.com/en-us/vsts/build-release/test/example-continuous-testing?view=vsts). Retrieved from <https://docs.microsoft.com/en-us/vsts/build-release/test/example-continuous-testing?view=vsts>
 - Short tutorials, quick starts, and resources that introduce the various concepts and tasks of VSTS Build and Release.

- Microsoft. (n.d.). [Build and release - Videos](https://docs.microsoft.com/en-us/vsts/pipelines/?view=vsts#videos). Retrieved from <https://docs.microsoft.com/en-us/vsts/pipelines/?view=vsts#videos>

Video

View the following videos to learn how to manage releases using VSTS and the Azure portal:

- Microsoft Visual Studio. (2016). [Release Management – Azure Deployments](#) [Video] | [Transcript](#). Retrieved from <https://www.youtube.com/watch?v=fd3WLghC1y0> (9:44).
- Microsoft Visual Studio. (2017). [Build a CI/CD pipeline from Visual Studio Team Services](#) [Video] | [Transcript](#). Retrieved from <https://www.youtube.com/watch?v=yNESSInONyk> (18:00).

u09s1 - Learning Components

- Create manual and automatic deployment of releases to Azure
- Navigate VSTS menus and screens.
- Push and verify changes from VS to VSTS
- Create manual and automatic VSTS builds.
- Create Azure web apps.
- Use VS IDE text editors.
- Write Git commit changes of code.

u09d1 - Define VSTS Release Definitions, Releases, and the Relationship Between Them

DevOps Continuous Delivery (CD) is implemented in VSTS via release definitions from which releases are generated. Understanding the meaning of these VSTS entities and how they relate to each other is vital for successful DevOps using VSTS.

In this discussion, you will go over VSTS release definitions, VSTS releases, and the relationship between these two entities.

Discuss the following:

1. What is meant by VSTS a release definition, the components of a release definition, and how you can create a release definition.
2. What is meant by a VSTS release and how you can create a release.
3. The relationship between VSTS release definitions and VSTS releases and represent this relationship as UML class diagram.

As with all discussions in this course, it is recommended that you make your initial post by 12 a.m. Central Standard Time on Wednesday evening to allow time for your peers to respond.

Response Guidelines

The expectation within the course discussions is to respond to at least two posts by each Sunday evening, but it is highly recommended that you extend the dialog further. Responding over multiple days will help in stimulating a lively discussion. Start by submitting your initial post on (or before) Wednesday. If you provide responses to your peers on Thursday, Saturday, and Sunday, and your peers reciprocate with their responses, you will have more opportunity for in-depth interaction with your classmates and the instructor.

Prior to Sunday evening, read your peers' posts, and provide feedback to at least two of them.

Course Resources

Undergraduate Discussion Participation Scoring Guide

u09d1 - Learning Components

- Create manual and automatic deployment of releases to Azure
- Navigate VSTS menus and screens.
- Push and verify changes from VS to VSTS
- Create manual and automatic VSTS builds.
- Create Azure web apps.
- Use VS IDE text editors.
- Write Git commit changes of code.

u09a1 - Manage ASP.NET Core CD Releases for Azure Deployment Using VS, VSTS, and Azure Portal

Overview

To manage releases quickly and efficiently, one must first set up a complete DevOps CI/CD pipeline. A complete DevOps CI/CD pipeline ensures automated and controlled updates to software systems. VS, VSTS, and the Azure portal provide all the tools you need to setup and configure a CI/CD pipeline. Skillfully and efficiently using these tools will ensure smooth DevOps in your organization.

In this assignment, you manage the release of an ASP.NET core application for Azure deployment using VS, VSTS, and the Azure portal. You configure a complete CI/CD pipeline using these tools. After making code changes to the VS project, your CI/CD pipeline will automatically deploy the changed application to Azure. You will also explain how you used the VS, VSTS, and the Azure portal to implement and validate the CI/CD pipeline.

Directions

Follow the directions in the Unit 9 Assignment Directions document found in the Resources.

Submission Requirements

Complete and submit all work specified in the directions using the Assignment Submission Template. Make sure to rename it using the appropriate unit number.

Course Resources

Assignment Submission Template

Unit 9 Assignment Instructions

Unit 10 >> Controlling Application CD Deployment with Approvals & Gates

Introduction

Unit 10 discusses release approval in VSTS. You will learn about manual approval and automatic approvals (gates). There is no assignment for Unit 10.

Learning Activities

u10s1 - Studies

Reading

Read or view the following Internet resources to learn about release approvals and its practices in VSTS:

- Microsoft Developer. (2018). [Taking control over your releases with VSTS Release Management \(RM\) Release Gates](#). [Video] | [Transcript](#). Retrieved from https://www.youtube.com/watch?v=7WLCqwhTZ_4 (5:28).
- Microsoft. (2018). [Release approvals and gates overview](#). Retrieved from <https://docs.microsoft.com/en-us/vsts/pipelines/release/approvals/?view=vsts>
- Microsoft. (2018). [Release deployment control using approvals](#). Retrieved from <https://docs.microsoft.com/en-us/vsts/pipelines/release/approvals/approvals?view=vsts>
- Microsoft. (2018). [Release deployment control using gates](#). Retrieved from <https://docs.microsoft.com/en-us/vsts/pipelines/release/approvals/gates?view=vsts>

Read the following Skillsoft resource:

- Olausson, M., & Ehn, J. (2015). [Continuous Delivery with Visual Studio ALM 2015](#) (1st ed.). Apress.
 - Chapter 10: "Building a Deployment Pipeline," pp. 306–334.

u10s1 - Learning Components

- Create manual and automatic deployment of releases to Azure
- Navigate VSTS menus and screens.
- Push and verify changes from VS to VSTS
- Create manual and automatic VSTS builds.

u10d1 - Complete Release Pre-deployment Approval Using VS, VSTS, and Azure Portal.

VSTS release management contains controls for approval of releases before they are deployed to their target environments (pre-deployment) and after they have been deployed (post-deployment). These approvals can be manual or can be automated via gates. These approval controls ensure that no release is deployed unless it has been approved by the stakeholders of the deployment process and thus ensuring proper management of the release.

In this discussion, you configure a pre-deployment approval on a release in an automated VSTS pipeline. Another person will approve the release so that it gets deployed automatically to Azure.

Note: because you need a peer to approve your release and provide a screenshot, plan to make your initial posts earlier than usual this week to allow time to send and receive these screenshots.

Complete these steps:

1. Create an automated DevOps pipeline for an ASP.NET Core web application using VS Continuous Delivery Tools for Visual Studio extension.
2. Edit the generated release definition to configure a pre-deployment approval for an approver different from yourself using VSTS.
3. Commit and push changes to the Pages/About.cshtml page using VS. The changes will result in a new build, but the release will be pending approval.
4. Ask the approver to approve the release of your application.
5. Once the approver approved the release, verify that the release was actually deployed to Azure.

Provide four screenshots of your result of your work. The first screenshot should show the pre-deployment condition (approval) configuration in the release definition. The second screenshot should show the release is pending pre-deployment approval after you committed changes to the Page/About.cshtml page. The third screenshot should (ask the approver for this screenshot) should show the request to approve a release on the approver page. The fourth screenshot should show the second deployment of your application succeeded after the approver approved the release. See the Discussion u10d1 Screenshots file in the Resources for examples.

As with all discussions in this course, it is recommended that you make your initial post by 12 a.m. Central Standard Time on Wednesday evening to allow time for your peers to respond.

Response Guidelines

The expectation within the course discussions is to respond to at least two posts by each Sunday evening, but it is highly recommended that you extend the dialog further. Responding over multiple days will help in stimulating a lively discussion. Start by submitting your initial post on (or before) Wednesday. If you provide responses to your peers on Thursday, Saturday, and Sunday, and your peers reciprocate with their responses, you will have more opportunity for in-depth interaction with your classmates and the instructor.

Prior to Sunday evening, read your peers' posts, and provide feedback to at least two of them.

Course Resources

[Undergraduate Discussion Participation Scoring Guide](#)

[Discussion u10d1 Screenshots](#)

u10d1 - Learning Components

- Create a new build definition.
- Navigate to VSTS Build & Release menu.
- Create a VSTS project with Git repo.
- Save and queue a new build.
- Create a new build definition.
- Select build tasks.
- Create environments for the release.
- Select release tasks and enable continuous delivery.