

Introduction to Hands-on-Labs for Building a Continuous Delivery Release Pipeline with Team Foundation Server



Table of Contents

- Objectives 2
- Contents of the HOLs 3
- Completing the Labs in Sequence 4
 - With the Brian Keller VM 4
 - Without the Brian Keller VM 4
- Check-Off List for the HOLs 5
 - Operating Systems 5
 - Tools 5
 - SDKs 6
 - Third-party Libraries 6
- Setup Information 6
 - Operating System Links 6
 - Windows 7 6
 - Windows Server 2008 R2 6
 - Windows 8 6
 - Windows Server 2012 7
 - Windows Azure 7
 - Tools Links 7

Visual Studio 2012.....	7
Team Foundation Server 2012.....	7
TFS Build Controller.....	7
Test Controller and Agents	7
Microsoft Test Manager.....	7
WiX Tool Set.....	8
Web Deploy.....	8
TFS Build Extension	8
SDKs.....	8
ALM Rangers Material.....	8
Standard IIS Configurations	9
Copyright.....	10

Objectives

This series of labs teaches you how to implement a continuous delivery release pipeline by using Team Foundation Server 2012. You begin by building a simple release pipeline that supports continuous integration, but has no other automation. Subsequent labs modify and extend the pipeline until it supports continuous delivery and includes features such as automated deployments and automated testing.

A good starting point is to read the guidance that the labs are based on, which is [Building a Release Pipeline with Team Foundation Server 2012](#). This guidance emphasizes three patterns that are critical to developing a continuous delivery pipeline and whose importance is reinforced by the hands-on labs (HOL). The patterns are:

- Orchestration
- Automation
- Monitoring and Metrics

Even as new technologies become available, the underlying patterns that shape your development efforts will remain the same.

This document is an introduction to the labs and discusses the following topics:

- Contents of the HOLs.
- Prerequisites for the HOLs.

- Where to find more information about the prerequisites.
 - ALM Rangers material
 - Standard IIS configurations.
-

Contents of the HOLs

One of the biggest challenges in learning about distributed architectures such as the one required for these labs is creating the infrastructure. The details of setting up a complex infrastructure can be a serious distraction that makes focusing on the actual purpose of the labs difficult. To remove this difficulty, the labs have two versions. Labs 1 through 5 assume that you are using the [Brian Keller VM](#) for your infrastructure.

Lab 6 is considered an advanced lab, and assumes you are not using the Brian Keller VM. This is because it shows you how to install the components of the Trey Research Windows Phone 8 app and how to deploy a Windows Communication Foundation (WCF) service to Windows Azure. Currently, the Brian Keller VM cannot support these features, so you must create the entire infrastructure yourself. Here is a more detailed description of the labs.

- **Lab 1: Starting Point.** In this lab you install the Trey Research application and create the initial version of the Trey Research release pipeline.
- **Lab 2: Orchestration.** This is a set of four labs. In the first lab you orchestrate the commit stage of the pipeline. In the second lab you orchestrate the rest of the pipeline stages. In the third lab you configure the pipeline. In the fourth lab you test the orchestration.
- **Lab 3: Automation.** This is a series of three labs. In the first lab you automate the deployment of the WCF service. In the second lab you automate the deployment of the WPF application. In the third lab you automate the tests. At the conclusion of the Automation HOL, you will have created a continuous delivery pipeline.
- **Lab 4: Monitoring and Metrics.** This is a set of two labs. In the first lab you learn how to monitor a continuous delivery pipeline. In the second lab you learn how to create custom reports in Microsoft Team Foundation Server (TFS) to track some of the key metrics that are used with continuous delivery projects.
- **Lab 5: Adding New Stages to the Pipeline.** In this lab you learn how to add new stages to the pipeline. New stages contain new tests that improve how you test the pipeline
- **Lab 6: Advanced.** This is a set of two labs. The first lab is similar to the Starting Point lab, but has additional sections that show you how to add the components for the Windows Phone 8 app and the WCF Windows Azure service. The second lab shows you how to automatically deploy the components for the Windows Phone 8 app.

Completing the Labs in Sequence

As was mentioned above, there are two ways to complete all the labs. Here are the two possible sequences.

With the Brian Keller VM

Here is the sequence you should follow if you are using the Brian Keller VM.

1. Introduction (this document)
 2. Lab01 – Starting Point
 3. Lab2.1 – Orchestrating the Commit Stage
 4. Lab 2.2 – Orchestrating the Remaining Stages
 5. Lab 2.3 – Configuring the Pipeline
 6. Lab 2.4 – Testing the Orchestration
 7. Lab 3.1 – Automating the Deployment of the WCF Service
 8. Lab 3.2 – Automating the Deployment of the WPF App
 9. Lab 3.3 – Running the Automated Tests
 10. Lab 4.1 – Monitoring the Continuous Delivery Pipeline
 11. Lab 4.2 – Metrics for Continuous Delivery in TFS
 12. Lab 5 – Adding New Stages to the Pipeline
-

Without the Brian Keller VM

Here is the sequence you should follow if you are not using the Brian Keller VM.

1. Introduction (this document)
2. Lab 6.1 – Advanced Starting Point
3. Lab 2.2 – Orchestrating the Remaining Stages
4. Lab 2.3 – Configuring the Pipeline
5. Lab 2.4 – Testing the Orchestration
6. Lab 3.1 – Automating the Deployment of the WCF Service
7. Lab 6.2 – Automating the Deployment of the Windows Phone 8 App
8. Lab 3.2 – Automating the Deployment of the WPF App
9. Lab 3.3 – Running the Automated Tests
10. Lab 4.1 – Monitoring the Continuous Delivery Pipeline

11. Lab 4.2 – Metrics for Continuous Delivery in TFS

12. Lab 5 – Adding New Stages to the Pipeline

Check-Off List for the HOLs

The following tables list the operating systems, tools, SDK,s and third-party libraries that you'll need to complete the labs. Most of them are included in the Brian Keller VM. Those that aren't are noted. To complete the advanced labs, you'll need to set up your own infrastructure that includes the items listed in the tables.

The labs are designed to give you options about the technologies and number of computers that you want to use. For example, you can deploy to a Windows Azure virtual machine (VM) to run an IIS service if you want, but you can also run the labs on either a local or remote IIS machine.

Operating Systems

Operating System	Brian Keller VM	Advanced Labs (Windows Phone 8)	Advanced Labs (Windows Azure)
Windows 7	Yes		Windows Azure account
Windows Server 2008 R2	Yes		Windows Azure account
Windows Server 2012		Yes	Windows Azure account
Windows 8		Yes	Windows Azure account

Tools

Tool	Brian Keller VM	Advanced Labs (Windows Phone 8)	Advanced Labs (Windows Azure)
Microsoft Visual Studio 2012	Yes	Required	Windows Azure account
Team Foundation Server 2012 (TFS)	Yes	Not applicable	Not applicable
TFS build controller	Yes	Not applicable	Not applicable
Microsoft Test Manager 2012 (MTM)	Yes	Not applicable	Not applicable
Test controller and agents	Yes	Not applicable	Not applicable
Microsoft Office Excel	Yes	Not applicable	Not applicable
WiX Tool Set	No, user downloads this tool in the second	Not applicable	Not applicable

	automation lab		
Web Deploy v.3.0	Yes	Not applicable	Required for IIS

SDKs

SDK	Brian Keller VM	Advanced Labs (Windows Phone 8)	Advanced Labs (Windows Azure)
Windows Phone SDK	No, but not needed	Required	Not applicable
Windows Azure SDK	No, but not needed	Not applicable	Required

Third-party Libraries

Library	Brian Keller VM	Advanced Labs (Windows Phone 8)	Advanced Labs (Windows Azure)
TFS Build Extensions	No, user downloads library in Starting Point lab	Not applicable	Not applicable

Setup Information

This section contains links where you can find more information about the items listed in the tables.

Operating System Links

These links give information about how to develop applications for each operating system.

Windows 7

The Windows 7 Training Kit for Developers includes presentations, hands-on labs, and demos designed to help you learn how to build applications that are compatible with Windows 7. For more information see the [Windows 7 Training Kit for Developers](#).

Windows Server 2008 R2

For more information about Windows 2008 R2, see [Windows Server 2008 R2](#).

Windows 8

The Windows Dev Center is a great place to start learning about Windows 8. For more information see [Dev Center — Windows Store apps](#).

Windows Server 2012

For more about Windows Server 2012, go to [Windows Server 2012](#)

Windows Azure

The Windows Azure team has created a great deal of documentation. To sign up for a Windows Azure subscription or a trial, see [Windows Azure Purchase Options](#). You will also need to use the Windows Management Portal so that you can configure your Windows Azure service namespaces. For more information about Windows Azure, go to the [product website](#), which has a link to the documentation. One tutorial that may interest you is [Create a Virtual Machine Running Windows Server](#).

Tools Links

The following links give information about the various tools.

Visual Studio 2012

Visual Studio is the central tool for developing code on the Microsoft platform. For more information, go to the [Visual Studio](#) website.

Team Foundation Server 2012

Team Foundation Server 2012 is the collaboration platform at the center of Microsoft's application lifecycle management solution. For more information, go to the [Team Foundation Server](#) website.

TFS Build Controller

To learn more about how to configure and manage your build controller see [Configuring and Managing Your Build System](#).

Test Controller and Agents

To learn more about test controllers and agents see [Setting Up Test Machines to Run Tests or Collect Data](#).

Microsoft Test Manager

MTM works with Team Foundation Server to coordinate the test controllers that run the manual and automated tests that run in the Lab Management environment. You use the MTM test plan management features to organize the tests for a particular build. For more information, see the following articles.

- [Setting Up Test Controllers in Lab Environments](#).
- [Getting Started with Lab Management](#).
- [Configuring and Administering Lab Management](#).
- [Creating Lab Environments](#).

- [Using a Lab Environment for Your Application Lifecycle](http://msdn.microsoft.com/en-us/library/dd997438.aspx). at <http://msdn.microsoft.com/en-us/library/dd997438.aspx>.
- [ALM Rangers' Visual Studio Lab Management Guide](#).

WiX Tool Set

WiX allows you to build Windows installation packages from XML source code. WiX supports completely automated installations and uninstallations, and is integrated with Visual Studio and MSBuild. For more information about the Windows Installer, see [Windows Installer](#). For more information about writing WiX files, refer to the [WiX tutorial](#) or the [WiX manual](#). You can download the WiX tool set from [CodePlex](#).

Web Deploy

[Web Deploy](#) contains the tools that transform the configuration files and package the files to be deployed.

TFS Build Extension

The [Community TFS Build Extensions](#) is a Microsoft ALM Ranger project on CodePlex that offers a number of useful workflow activities to help you construct a release pipeline. It is used in the orchestration labs.

SDKs

Here's where you can download the SDKs that are required to complete the labs.

- [Windows Phone 8 SDK](#)
 - [Windows Azure SDK](#)
-

ALM Rangers Material

The ALM Rangers have a great deal of material that you will find helpful as you create your release pipeline.

- [Visual Studio ALM Rangers Visual Studio Test Tooling Guidance](#).
 - [Visual Studio ALM Rangers Visual Studio Lab Management Guide](#).
 - [Visual Studio ALM Rangers Team Foundation Build Customization Guide](#).
 - [Visual Studio ALM Rangers, Visual Studio Team Foundation Server Branching and Merging Guide](#).
 - [Visual Studio ALM Rangers Visual Studio ALM Quick Reference Guidance](#).
 - [Visual Studio ALM Rangers Blog](#).
-

Standard IIS Configurations

This section is optional and is provided as a reference for when you set up the IIS websites in the Starting Point lab (both the standard and advanced versions). To ensure that you have the correct configurations, you can consult the [IIS Configuration Reference](#). You should also make sure that the WCF Service is **ON**.

This is a list of all the IIS services that need to be installed in order for the WCF service to work correctly.

Display Name	Name	Install State
-----	----	-----
[X] Application Server	Application-Server	Installed
[X] .NET Framework 4.5	AS-NET-Framework	Installed
[X] Web Server (IIS) Support	AS-Web-Support	Installed
[X] File And Storage Services	FileAndStorage-Services	Installed
[X] File and iSCSI Services	File-Services	Installed
[X] File Server	FS-FileServer	Installed
[X] Storage Services	Storage-Services	Installed
[X] Web Server (IIS)	Web-Server	Installed
[X] Web Server	Web-WebServer	Installed
[X] Common HTTP Features	Web-Common-Http	Installed
[X] Default Document	Web-Default-Doc	Installed
[X] Directory Browsing	Web-Dir-Browsing	Installed
[X] HTTP Errors	Web-Http-Errors	Installed
[X] Static Content	Web-Static-Content	Installed
[X] HTTP Redirection	Web-Http-Redirect	Installed
[X] Health and Diagnostics	Web-Health	Installed
[X] HTTP Logging	Web-Http-Logging	Installed
[X] Logging Tools	Web-Log-Libraries	Installed
[X] Request Monitor	Web-Request-Monitor	Installed
[X] Performance	Web-Performance	Installed
[X] Static Content Compression	Web-Stat-Compression	Installed
[X] Dynamic Content Compression	Web-Dyn-Compression	Installed
[X] Security	Web-Security	Installed
[X] Request Filtering	Web-Filtering	Installed
[X] Basic Authentication	Web-Basic-Auth	Installed
[X] Client Certificate Mapping Authentic...	Web-Client-Auth	Installed
[X] Digest Authentication	Web-Digest-Auth	Installed
[X] IIS Client Certificate Mapping Authe...	Web-Cert-Auth	Installed
[X] IP and Domain Restrictions	Web-IP-Security	Installed
[X] URL Authorization	Web-Url-Auth	Installed
[X] Windows Authentication	Web-Windows-Auth	Installed
[X] Application Development	Web-App-Dev	Installed
[X] .NET Extensibility 3.5	Web-Net-Ext	Installed
[X] .NET Extensibility 4.5	Web-Net-Ext45	Installed
[X] ASP.NET 3.5	Web-Asp-Net	Installed
[X] ASP.NET 4.5	Web-Asp-Net45	Installed
[X] ISAPI Extensions	Web-ISAPI-Ext	Installed
[X] ISAPI Filters	Web-ISAPI-Filter	Installed

[X] IIS Hostable Web Core	Web-WHC	Installed
[X] Management Tools	Web-Mgmt-Tools	Installed
[X] IIS Management Console	Web-Mgmt-Console	Installed
[X] IIS Management Scripts and Tools	Web-Scripting-Tools	Installed
[X] .NET Framework 3.5 Features	NET-Framework-Features	Installed
[X] .NET Framework 3.5 (includes .NET 2.0 and 3.0)	NET-Framework-Core	Installed
[X] HTTP Activation	NET-HTTP-Activation	Installed
[X] Non-HTTP Activation	NET-Non-HTTP-Activ	Installed
[X] .NET Framework 4.5 Features	NET-Framework-45-Fea...	Installed
[X] .NET Framework 4.5	NET-Framework-45-Core	Installed
[X] ASP.NET 4.5	NET-Framework-45-ASPNET	Installed
[X] WCF Services	NET-WCF-Services45	Installed
[X] HTTP Activation	NET-WCF-HTTP-Activat...	Installed
[X] Message Queuing (MSMQ) Activation	NET-WCF-MSMQ-Activat...	Installed
[X] Named Pipe Activation	NET-WCF-Pipe-Activat...	Installed
[X] TCP Activation	NET-WCF-TCP-Activati...	Installed
[X] TCP Port Sharing	NET-WCF-TCP-PortShar...	Installed
[X] Message Queuing	MSMQ	Installed
[X] Message Queuing Services	MSMQ-Services	Installed
[X] Message Queuing Server	MSMQ-Server	Installed
[X] Remote Differential Compression	RDC	Installed
[X] User Interfaces and Infrastructure	User-Interfaces-Infra	Installed
[X] Graphical Management Tools and Infrastructure	Server-Gui-Mgmt-Infra	Installed
[X] Server Graphical Shell	Server-Gui-Shell	Installed
[X] Windows PowerShell	PowerShellRoot	Installed
[X] Windows PowerShell 3.0	PowerShell	Installed
[X] Windows PowerShell 2.0 Engine	PowerShell-V2	Installed
[X] Windows PowerShell ISE	PowerShell-ISE	Installed
[X] Windows Process Activation Service	WAS	Installed
[X] Process Model	WAS-Process-Model	Installed
[X] .NET Environment 3.5	WAS-NET-Environment	Installed
[X] Configuration APIs	WAS-Config-APIs	Installed
[X] WoW64 Support	Wow64-Support	Installed

Copyright

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet website references, may change without notice. You bear the risk of using it. Some examples depicted herein are provided for illustration only and are fictitious. No real association or connection is intended or should be inferred.

© 2013 Microsoft. All rights reserved.

Microsoft, Windows, Windows Server, Windows Vista, Windows Azure, Windows PowerShell, Silverlight, Expression, Expression Blend, MSDN, IntelliSense, IntelliTrace, Internet Explorer, SQL Azure, SQL Server,

Visual C#, Visual C++, Visual Basic, and Visual Studio are trademarks of the Microsoft group of companies.

All other trademarks are the property of their respective owners.