entwickler.press
shortcuts

TFS 2012 Team Build

Architektur und Installation

Tobias Richling, Michael Klei

Tobias Richling, Michael Klei

TFS 2012 Team Build

Architektur und Installation

ISBN: 978-3-86802-474-6

Ein Imprint der Software & Support Media GmbH

1 Einleitung

Eine Software zu erstellen, geht weit über das Kompilieren im Visual Studio hinaus. Der Erstellungsprozess bezieht mehrere Lösungen mit ein und erzeugt Kompilate verschiedenen Typs, zum Beispiel Clientapplikationen, Dienste und ein Setup. Im Zuge der Erstellung müssen automatisierte Tests ausgeführt werden, die Änderungen seit dem letzten Build ermittelt und bei erfolgreicher Ausführung die fertigen Dateien für den Tester oder die Auslieferung bereitgelegt werden. Für verschiedene Codelinien wie Entwicklung, Main oder die Releases müssen verschiedentlich getaktete Builds definiert werden.

In diesem Kapitel

- werden ausgehend von einem Branch-Modell Build-Definitionen erzeugt
- werden private Builds erläutert
- wird der dem Team Build zugrunde liegende Workflow erläutert
- wird vorgestellt, wie man diesen Workflow mit eigenen Aktivitäten anpassen kann
- wird demonstriert, wie man auf Build-Ereignisse reagieren kann

Voraussetzungen für dieses Kapitel:

- eine vorhandene TFS-Installation
- ein einfaches Branch-Modell

2 Grundlagen

2.1 Architektur

Der Team Build ist als Bestandteil des TFS natürlich in die Gesamtarchitektur eingebunden. Dementsprechend benötigt er einen TFS und dessen Datenbanken, um grundsätzlich zu funktionieren. Darüber hinaus gibt es zwei wesentliche logische Komponenten, die den Team Build ausmachen: der Build-Controller und der oder die Build-Agenten. Beide sind Bestandteile des Build Service.

Logische Architektur

Der Build Service ist die logische Einheit, die die Grundlage für Controller und Agenten darstellt. Der Build-Dienst muss auf jedem Rechner installiert sein, auf dem ein Controller und/oder ein Agent ausgeführt werden soll.

Der Build Controller nimmt alle Build-Anforderungen entgegen, legt die Ausführungsreihenfolge fest und startet die Ausführung der Builds. Auf einem physikalischen Rechner kann nur jeweils ein Build Controller installiert werden, der immer einer Teamprojektsammlung zugeordnet ist. Die Abarbeitung eines Build-Laufs beginnt auf dem Controller, auf dem Initialisierungsarbeiten erledigt werden. Die Hauptarbeit für die Durchführung der Builds delegiert der Controller an einen Build Agent. Die Arbeit des Build Controllers erfordert nicht viel CPU-Leistung, kann aber speicherintensiv sein – also lieber mehr Speicher als leistungsstarker Prozessor (Abbildung 2.1).

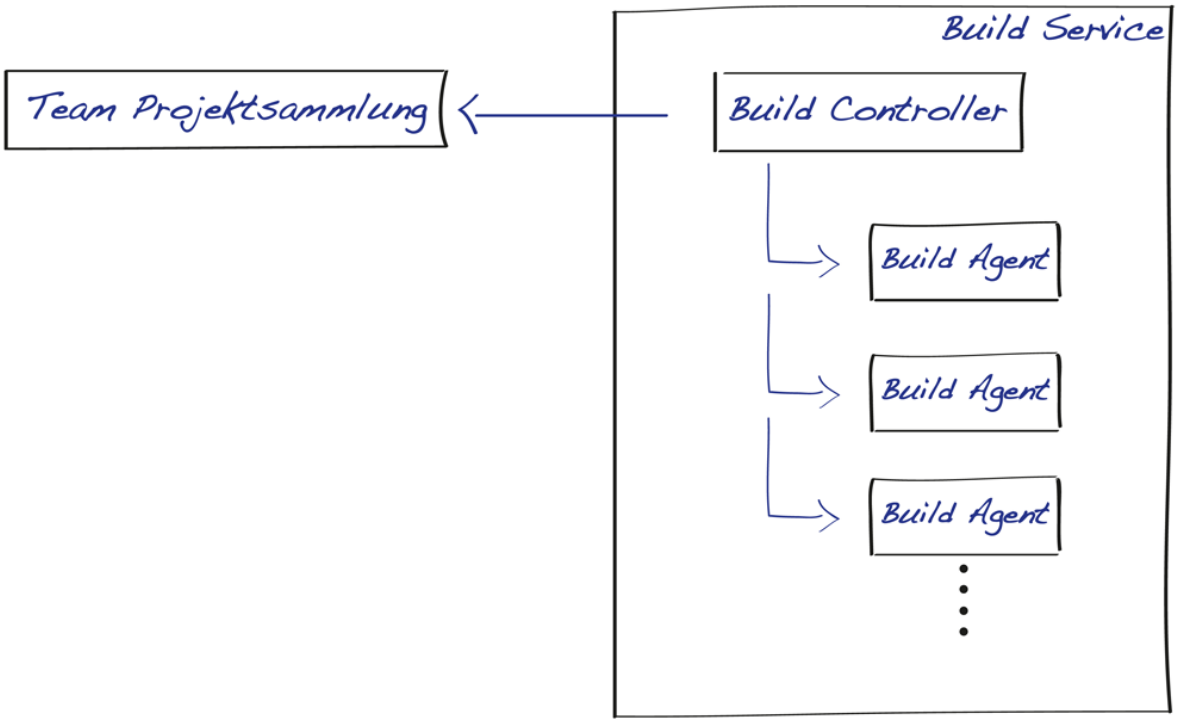

Abbildung 2.1: Logische Team-Build-Architektur

Der Build Agent führt die Hauptarbeit bei der Ausführung von Builds aus. Jeder Build Agent ist einem Build Controller zugeordnet, der für ihn verantwortlich ist um ihm die Ausführung eines konkreten Builds zuweist. Während es pro Teamprojektsammlung genau einen Build Controller geben kann, können mehrere Build Agents existieren. Jeder Build Agent kann zur gleichen Zeit nur einen Build ausführen. Wenn mehr Builds angefordert werden als Build Agents verfügbar sind, werden diese vom Controller in einer Warteschlange gesammelt. Sie können auf dem gleichen Rechner wie der Build Controller laufen, oder auf separaten Maschinen. Auf einem physikalischen Computer können auch mehrere Build Agents ausgeführt werden. Da der Build Agent die rechenzeitintensiven und datenlastigen Tätigkeiten im Team Build übernimmt, sollten nicht zu viele Agents auf einem Rechner laufen. Wenn man mit der Einrichtung einer Build-

Umgebung beginnt, kann man mit einer kleinen Anzahl an Agents beginnen – für den Anfang tut es für gewöhnlich ein einziger. Später wird die Anzahl der durchzuführenden Builds eventuell steigen, dann kann man weitere Agents ins System bringen. Es ist auch möglich, Agents für spezielle Aufgaben einzurichten, zum Beispiel einen Agents, der nur nächtliche Builds durchführt, oder eine Gruppe von Agents, die nur Continuous Integration Builds durchführen. Zu diesem Zweck kann man Agents Tags zuweisen. Bei der Definition eines Builds können ebenfalls Tags angegeben werden. Der Build Controller versucht dann, einen freien Agent zu finden, der mit den angegebenen Tags übereinstimmt.

Physikalische Architektur

Es gibt verschiedene Möglichkeiten, die logischen Komponenten auf physikalische Computer zu verteilen. Alle gezeigten Ansätze gehen von einer Teamprojektsammlung aus, für jede weitere Sammlung wird jeweils ein weiterer Build Controller fällig, der auf einem eigenen Rechner laufen muss und somit wiederum mit jedem Architekturansatz kombiniert werden kann. Drei mögliche Architekturansätze sind in Abbildung 2.2 zusammengefasst.

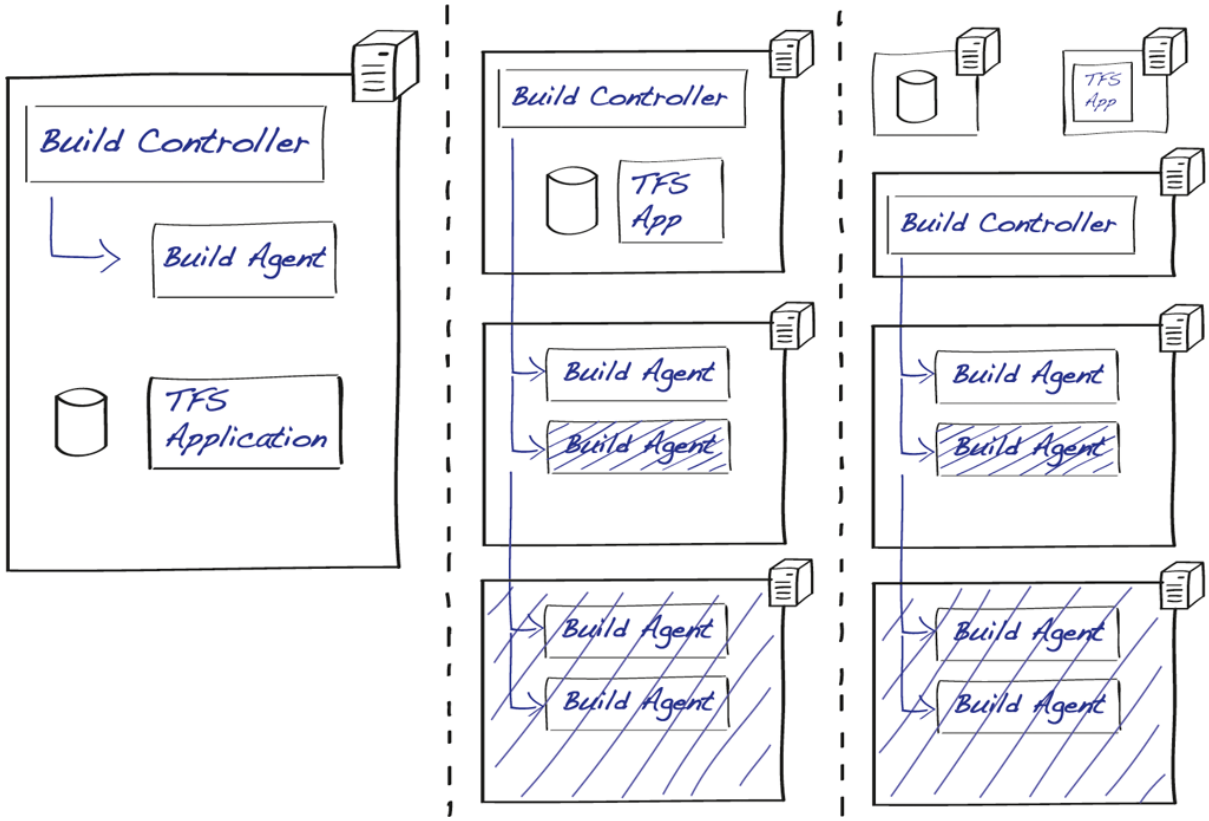

Abbildung 2.2: Drei mögliche physikalische Team-Build-Architekturen

Die einfachste Topologie ist die, alle Komponenten auf einer Maschine zu installieren (in der Abbildung ganz links). Die Vorteile dieser Variante liegen auf der Hand: Sie ist einfach zu installieren und zu warten, und es fallen relativ geringe Hardwarekosten an. Für Einzelentwickler lässt sich dieses Setup auf dem Notebook einrichten und betreiben. Für kleine Teams reicht ein PC mit normaler Desktop-Hardware. Abhängig von der Anzahl der Builds, Teamprojekte und Entwickler kann man die Rechnerleistung langsam steigern. Wenn man leistungsfähige Server-Hardware einsetzt, kann man bis zu 15 Entwickler mit dieser Installation bedienen. Dieses Deployment läuft auch auf virtuellen Maschinen sehr gut.

Die erforderliche Rechenleistung hängt wesentlich von der Anzahl und dem Umfang der durchzuführenden Builds ab. Die Versionskontrolle hat relativ moderate Anforderungen, mit

durchschnittlicher Hardware kann man auch größere Teams bedienen. Solange nur ein Build parallel ausgeführt werden muss, reicht auch ein einzelner Rechner aus. Es kann aber leicht zu einer Situation kommen, wo mehrere Builds parallel laufen sollen. Dies ist nur möglich, wenn auch mehrere Build Agents vorhanden sind. Und dann wird es auf einer Maschine schnell eng. In diesem Fall kann man das erste Szenario erweitern, indem man einen weiteren Rechner aufsetzt und dort weitere Build Agents installiert (in der Abbildung mittig). Es ist für durchschnittliche Deployments problemlos möglich, den Build Controller auf einem Rechner mit den übrigen TFS-Diensten laufen zu lassen und nur die Agents auf weitere Knoten zu verteilen. Natürlich kann der Build Controller auch auf einem separaten Rechner installiert werden.

Sollte auch dieser Umfang nicht mehr ausreichen, kann man auf die große Variante zurückgreifen und alle Komponenten separat installieren (in der Abbildung links). Bei dieser Größenordnung wird die TFS-Datenbank ebenfalls auf einem eigenen dedizierten Datenbankserver untergebracht, das Application Tier bekommt einen eigenen Rechner, ebenso der Build Controller. Je nach Leistung der einzelnen Build-Knoten können dort ein oder mehrere Agents laufen.

2.2 Installation und Konfiguration

Einrichten eines Computers als Build-Computer

Damit ein Computer als Build-Computer fungieren kann, müssen die entsprechenden Komponenten installiert werden. Im Wesentlichen handelt es sich dabei um den Team Foundation Build Service, der sich in dem Prozess *TFSBuildServiceHost.exe* manifestiert und in der Regel als Windows-Dienst ausgeführt wird. Die Installation kann auf einem Rechner erfolgen, auf dem keine weiteren TFS-Komponenten installiert sind. In diesem Fall handelt es sich um eine reine Build-Maschine, die separat von der Datenbank und

den restlichen Diensten des TFS läuft. Die andere Möglichkeit, für kleinere Teams zu empfehlen, ist es, den Build-Dienst auf dem gleichen Rechner mit den übrigen Komponenten des TFS zu installieren – in diesem Fall bietet es sich an, dies direkt im Anschluss an die Installation zu erledigen. In diesem Fall fährt man in dem Dialog aus Abbildung 2.3 einfach mit dem Punkt *Configure Team Foundation Build Service* fort.

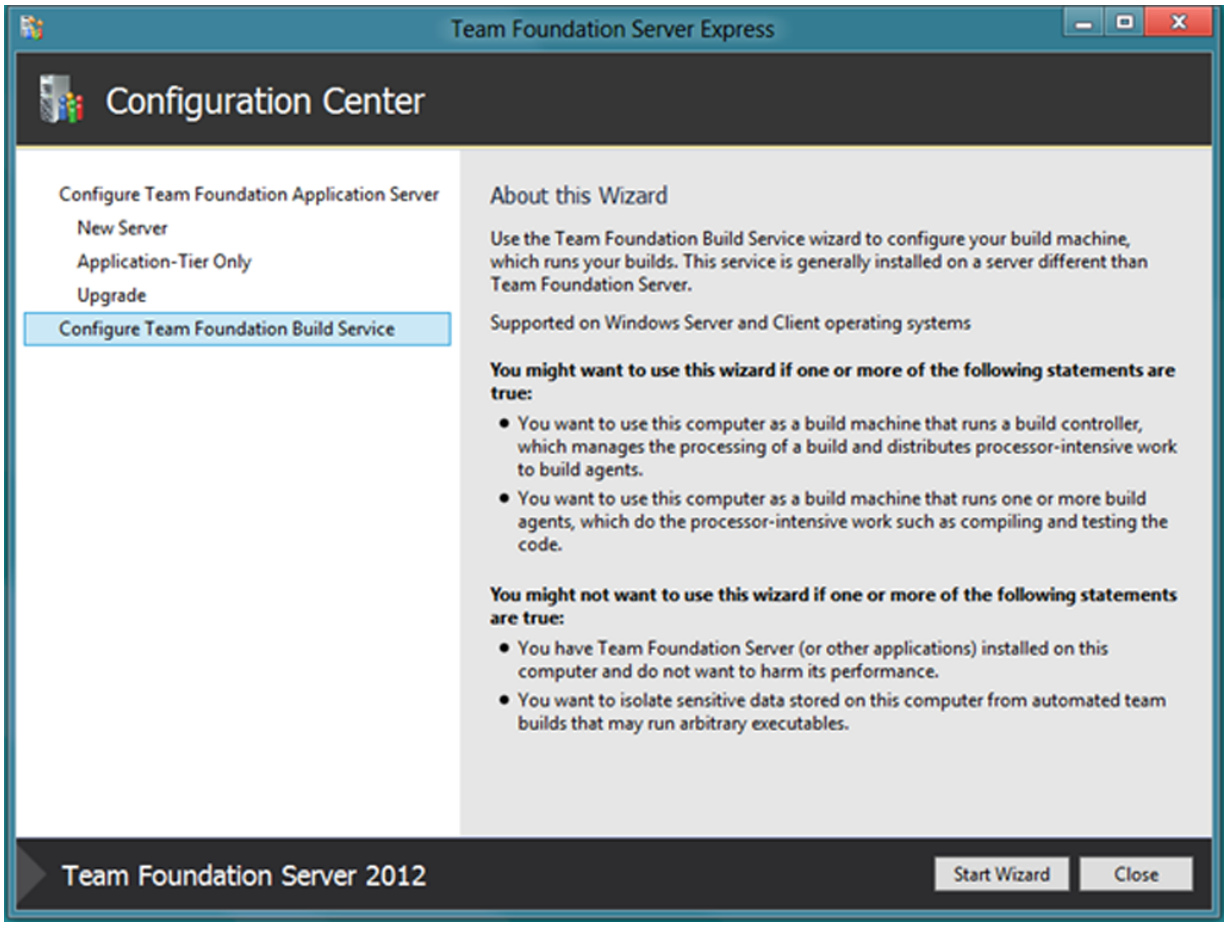

Abbildung 2.3: Assistent zur Installation der Build-Dienste

Die Installation erfolgt in drei einfachen Schritten. Im ersten Schritt muss eine Teamprojektsammlung angegeben werden, für die der zu installierende Build Controller tätig sein soll. Ein Build Controller kann immer nur für genau ein Teamprojekt zuständig sein, und jede Teamprojektsammlung hat genau einen Build Controller. Auf einer Maschine kann nur genau ein Build Controller installiert werden. Da für jede