International Journal of Innovation Scientific Research and Review
Vol. 02, Issue, 10, pp.445-449, October, 2020
Available online at http://www.journalijisr.com

Research Article

ISSN: 2582-6131

AUTOMATING SOFTWARE CODE DEPLOYMENT USING CONTINUOUS INTEGRATION AND CONTINUOUS DELIVERY PIPELINE FOR BUSINESS INTELLIGENCE SOLUTIONS

*** Farhana Sethi**

Global Data & Analytics Business Intelligence - Quality & Governance Manager with Schlumberger Oilfield, Texas, Houston since 2006.

Received 15th August 2020; Accepted 20th September 2020; Published online 24th October 2020

ABSTRACT

Continuous Integration (CI) is an approach to be continually validating the state of a codebase through automated testing. Best achieved through integration with version control. Continues Delivery/Deployment (CD) is a method to regularly deploy artifacts that successfully permit the CI phase to ensure assurance around deployment. Continues integration, continuous Deliver/Deployment are like the vectors that have the same direction, but different magnitude. Objectives of CI CD for any software development environment is to make code development and code release Faster in robust fashion. CI CD can automatically prepare and track a release to production, eliminate manual tasks and developers become more productive and eventually quality will not be compromised as standards and governance can be practices on central infrastructure.

Keywords: Database Management, Security, integrity, and protection, Database design, modeling and Management, Access methods.

INTRODUCTION

Modern Information Technology (IT) practices are heavily dependent on the organizations' ability moving changes/features/fixes live in an agile and errorless way. As technology evolves, the requirement to implement automation practices increases exponentially. Over the past decades, waterfall software development practices are substituted by agile methods of software development. The Continuous Integration (CI) and Continues Delivery/Deployment (CD) or collectively named as CI/CD is a subset of a method drive from DevOPS. DevOPS is a set of practices that combines software development (Dev) and IT operations (Ops). It aims to shorten the systems development life cycle and provide continuous delivery with high software quality. DevOps is complementary with Agile software development; several DevOps aspects came from Agile scrum methodology. Agile scrum methodology is a project management system that relies on incremental development. Each iteration consists of two- to four-week sprints, where each sprint's goal is to build the most important features first and come out with a potentially deliverable product. While Agile, CI/CD, and DevOps are not the same, they provision one another. Agile emphases on the development process, CI/CD on practices, and DevOps on the philosophy.

In this paper, our focus is on CI/CD for Business Intelligence solutions in Software Code Development environment.

Software Development Life Cycle (SDLC) is a standard business practice to building any software applications. It has Seven major stages:
1) Planning and Requirement gathering 2) Define Requirement 3) Design and Prototyping; 4) Implementation or coding; 5) Testing; 6) Deployment; 7) Maintenance.
CI/CD methods are applicable from stage 4 to 6. As Figure 1 describes how CI and CD integrate in each phase of code development and deployment.

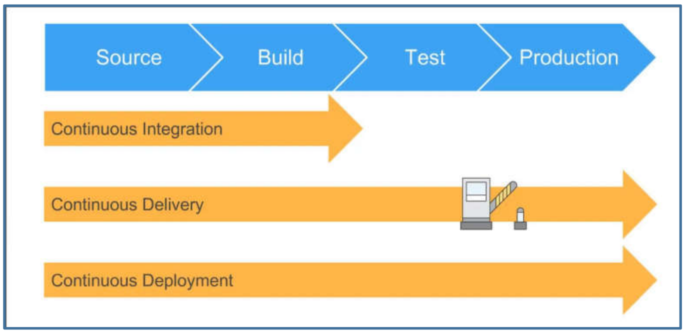

Figure 1: High level Overview of CI/CD in Code Development environment

FOUNDATIONS

Background

Continuous practices, i.e., continuous integration, delivery, and deployment, are the software development industry practices that enable organizations to frequently and reliably release new features and products. This has become one of the standard deployment strategy using agile methods for software engineering. Here we give an overview of continuous software engineering (e.g., continuous integration, continuous delivery, and continuous deployment) paradigm. 1"Continuous software engineering is an emerging area of research and practice. It refers to develop, deploy and get quick feedback from software and customer in a very rapid cycle. Continuous software engineering involves three phases: Business Strategy and Planning, Development and Operations. This study focuses on only three software development activities: continuous integration, continuous delivery and continuous deployment"1

Continuous Integration (CI) is the first part of the Continuous delivery pipeline. It is the procedure of automating the build and testing of code every stage a developer commits changes to source code repository.

***Corresponding Author: Farhana Sethi,**
Global Data & Analytics Business Intelligence - Quality & Governance Manager with Schlumberger Oilfield, Texas, Houston since 2006.

Continuous delivery (CD) is the execution of a pipeline (workflow) where code is regularly checked in, tests are programmed, and deployments are recurrent. This is key to a truly agile process, where release-ready increments are created during every sprint. The aim of Continuous Delivery is to keep the solution in a deploy-able state at any given time. This does not present that project is completely finished, however the feature collections that are offered are examined, tested, fixed and set to deploy.

Figure 2 describes the Overview of CI and CD in Code Development stages from code development to deployment.

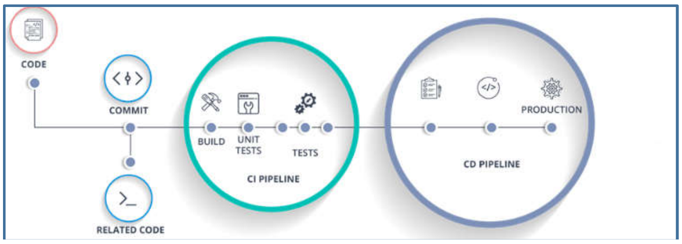

Figure 2. Overview of CI and CD in Code Development stages

RESEARCH SCOPE

The scope of this research paper is to apply CI/CD methods for Business Intelligence solutions using Microsoft Cloud environment and technology required for Data and Analytics solutions. We have considered four major components that are the key elements of any Business Intelligence solutions.

Database Deployments (Azure SQL Data warehouse) Azure SQL Data Warehouse is a managed petabyte-scale service with controls to manage compute and storage independently.

Scope of automation:

- Creating Database objects by deploying the SQL scripts to the target Data warehouse
- Create Schema and roles
- Provide necessary grants to roles
- Create Tables, Create Views and Stored Procedures

Azure Analysis Services is a fully managed platform as a service (PaaS) that provides enterprise-grade data models in the cloud. Use advanced mashup and modeling features to combine data from multiple data sources, define metrics, and secure your data in a single, trusted tabular semantic data model.

Scope of automation:

- Tabular Model Deployment; Refresh the Model
- Analysis Services Tabular Model Deployment

Azure Data Factory is Azure's cloud ETL service for scale-out server less data integration and data transformation. It offers a code-free UI for intuitive authoring and single-pane-of-glass monitoring and management.

Scope of automation

- Deploying the objects to the target Data Factory
- Linked Services; Datasets; Data Pipelines; Triggers

Reports visualization (Power BI). Power BI is a business analytics service by Microsoft. It aims to provide interactive visualizations and business intelligence capabilities with an interface simple enough for end users to create their own reports and dashboards.

Scope of automation

- Deploying and uploading the Power BI Reports;
- Creating Workspaces
- Adding Users/access rights to the Workspace
- Changing Dataset connection details

RESEARCH METHOD

In this research, we have used Azure DevOPS for Code management. Azure DevOps is a Software as a service (SaaS) platform from Microsoft that provides an end-to-end DevOps tool chain for developing and deploying software , as described in Figure 3 , the overall CI/CD Flow in Azure DevOps.

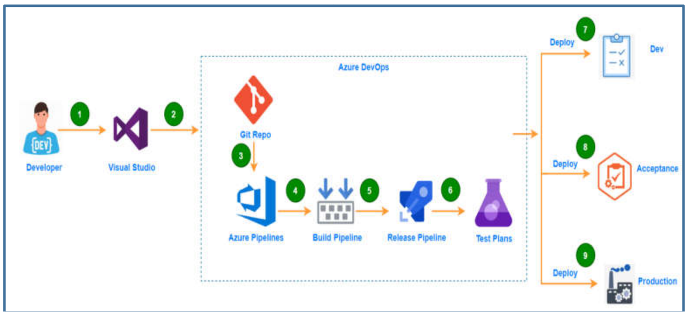

Figure 3. CI/CD Flow in Azure DevOps

We have used Visual studio for code development. Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs for Microsoft Windows, as well as web sites, web applications and web services.

Prerequisites:

A self-hosted build agent is required where CI CD pipeline will run automatically in cloud environment. It is requirement to build developer's code or deploy the software using Azure Pipelines, At minimum one agent is required, as we will add more code and developer to work, and we will ultimately need additional. An agent is installable software that runs one execution at a time at the event of pipeline run. A self-hosted agent can be any computer with Windows Operating System. Appendix 1 described the detail for build agent setup.

Approach

Code Development environment integrates with Azure DevOps Git repository (A Git repository is the. git/ folder inside a project for source Code management) to enable source control management and CI/CD. With source control, developers can collaborate, track and save their changes to a branch and these changes will be merged into the main branch and deployed to the higher-level environments (From development to Quality Assurance/User Acceptance Test environment to Production), where it will also be tested and validated. Once the above mentioned steps are complete the developed codes be published into the development environment as described in Figure 4.

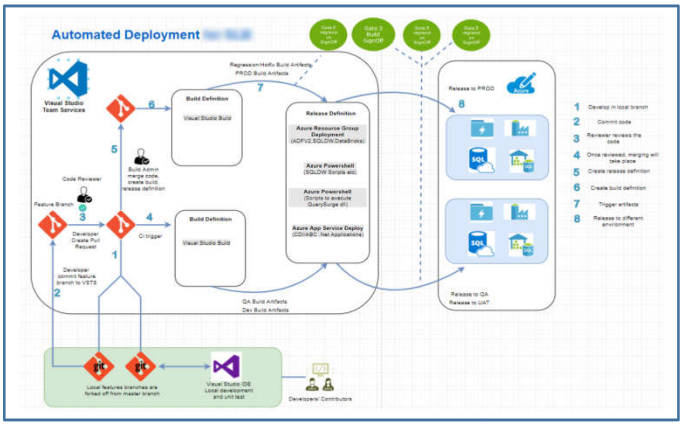
Automated Deployment
Gate 3 Build SignOff
Regression/Hotfix Build Artifacts
PROD Build Artifacts
Visual Studio Team Services
Build Definition
Visual Studio Build
Release Definition
Azure Resource Group Deployment (ADFV2,SQLDW,DataBricks)
Azure Powershell (SQLDW Scripts etc)
Azure Powershell (Scripts to execute QuerySurge dll)
Azure App Service Deploy (CDI/ABC .Net Applications)
Release to PROD
Build Admin merge code, create build, release definition
Code Reviewer
Feature Branch
Developer Create Pull Request
CI trigger
Developer commit feature branch to VSTS
QA Build Artifacts
Dev Build Artifacts
Release to QA
Release to UAT
1 Develop in local branch
2 Commit code
3 Reviewer reviews the code
4 Once reviewed, merging will take place
5 Create release definition
6 Create build definition
7 Trigger artifacts
8 Release to different environment
Local features branches are forked off from master branch
Visual Studio IDE Local development and unit test
Developers/ Contributors

Figure 4. Automated deployment using Azure DevOps (Git) Repository

CI AND CD PRACTICES IN AN INTEGRATED MANNER

Here we have explained how the sequence of Events for Solution deployment Flow in integrated manner.

Azure Data Factory Deployment

Each developer will create a feature branch, update and save their Data pipelines and activities. Once development is complete for a specific feature, the developer will save the changes and submit a pull request from feature branch to merge the changes into the master branch. The reviewer will review the changes and approve or reject the request. The reviewer can see all the files that have been changed as part of this release. Once all the steps are complete, the ADF Data pipeline can be published from master branch to the development environment. The ARM template for the Data pipeline and the Azure Resource Manager (ARM) template for the parameter file will be used for deploying the ADF to different environment. ARM Templates are a way to declare the objects in a JSON file which can be checked into source control and coped like any other code file.

Azure Data warehouse deployment

CI CD for Azure Data warehouse deployment is enabled using Data Tier Application Package (DACPAC).It is a single Windows file with a DACPAC extension. The file supports various XML format that contains database model, i.e. database objects in a binary format of database project compatible with SQL Server Data Tools (SSDT). SSDT is a modern development application for building relational databases. SQL Server Database project has to be created to include all the table, views, stored procedure and database scripts, when the Database project is build, visual studio generates a DACPAC file that can be used to automate the database deployment using Azure DevOps Release Pipeline.

Azure Analysis Service deployment

Azure Analysis Service extension can be added as a task to the Release Definition to deploy the model to the AAS. This task can also provide an option to configure the source database to which the connection needs to be established. Few deployment can also be achieved using Power Shell script. A schedule Azure Function is written to do the model data refresh on a scheduled frequency. Service Principal is used as authentication method to connect from Release Definition to Azure Analysis Service, since it works non-interactively within Azure Pipelines. In our research, we have enabled Tabular editor for AAS deployment. Tabular Editor provides a native hierarchical view of every object in the Tabular Model metadata. It is an Offline editor experience for SQL Server 2016 +/Azure Analysis Services tabular Models, a Lightweight application written in C# and it allows Advanced Scripting to automate manual tasks. It also allows integration with Power BI using XMLA Endpoint. Tabular Editor is a Command Line interface tool with highly productive means and can Integration with Azure DevOps.

Automated Quality testing

Quality checks and data validation (automatically) is the key for any successful release of a Business Analytic solution. In our research, we have used the automated testing and validation using Eggplant testing tool. It provides developers and testers with software to produce, program and execute dynamic quality testing. We used Azure DevOps Test Plans feature to create and integrate test plan with the CI CD pipeline as described in Figure 5.

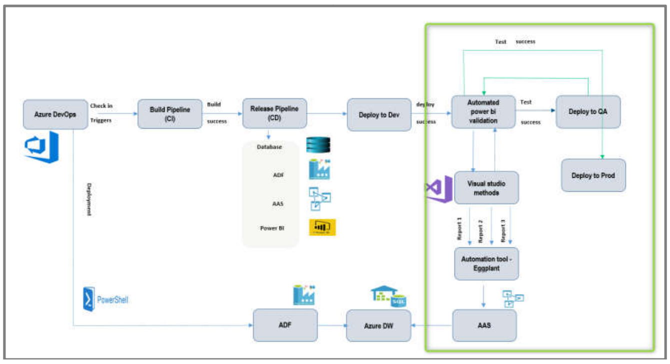

Figure 5. Automated testing integration with CI CD

Power BI Deployment

Power BI deployment is the last deployment component after quality testing as the Power BI report, once released, will be available to business users and this defines the success of the delivery to the business and customers. We used Deployment pipeline to deploy Power BI reports. Deployment Pipelines is an efficient and reusable tool (Provided by Microsoft Power BI) that enables BI creators to manage the lifecycle of organizational content. To be able to use the Deployment Pipeline Preview Feature the following conditions should be met: The User Should be a Pro User; The Organization should have at least one Premium Capacity; The User Should be an admin of a new Workspace Experience; The User is added as admin to an existing pipeline. CD Pipeline can be created per Team, if Continuous Delivery option is not enabled (i.e.: whenever there is a check-in and the CI is successful, there will be an immediate trigger for deployment in the CD Pipeline). In order to avoid the steps of creating the CD from scratch, Cloning feature will be used. If the CD is automated, then there will be a corresponding CD for every CI. In order to avoid the steps of creating the CD from scratch, Cloning feature will be used.

Azure Release Pipeline Approval/Pre Deployment Conditions

Approvals and gates gives further control over the execution and success of the deployment pipeline. Each step in a release pipeline can be built with pre-deployment and post-deployment settings that can embrace waiting for authority users to manually approve or reject deployments, and testing and validating the quality checks with any computerized systems until definite requirements are complete. Also, we can constitute a manual interference to pause the deployment pipeline and prompt developers to carry out manual tasks, then resume or reject the deployment. Once the approvals are in place the Release would be promoted to higher environments and finally deployed in Production. Assigned approvers would receive email notification along with the link to approve. Unless approved deployment will not take place as described in Figure 6.

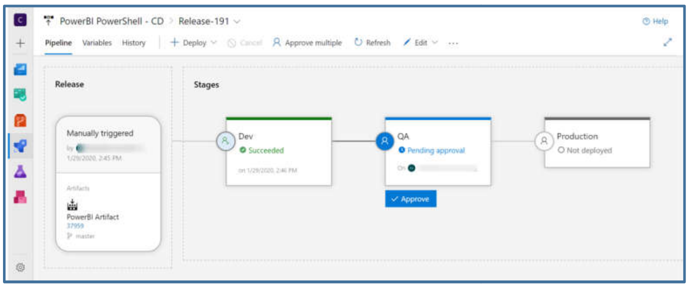

Figure 6: Azure Release Pipeline Approval for deployment to higher environment

Once it is approve and release, pipeline shows with success as described in Figure 7.

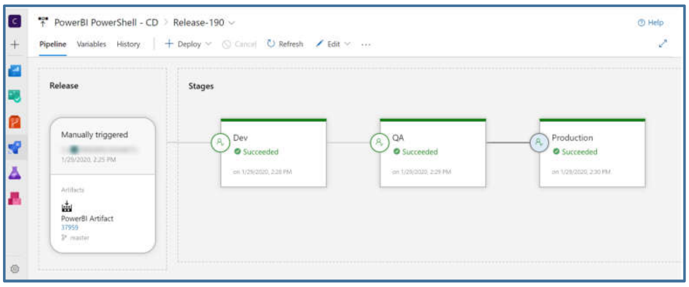

Figure 7: successful Azure Release Pipeline

CONCLUSIONS

Continuous Integration increase visibility, reduce overhead, increase consistency, inspire assurance and mitigate risk. However, Continuous delivery makes software release processes as efficient and repeatable as possible. In this paper, we demonstrate that how Continuous Integration and Continuous Delivery Pipelines can be implemented for Business intelligence solutions. The proposed methods can significantly reduce the time and gain efficiency for software release management. The developed scripts can be implemented in real time or near real time in any software development environment by the extension of similar approach.

Nomenclature

CI/CID: Continuous Integration (CI) and Continues Delivery/Deployment (CD);ADF: Azure Data factory; AAS: Azure Analysis Service; ADW: Azure Data warehouse ;DACPC: Data Tier Application Package; SSDT: SQL Server Data Tools; ARM: Azure Resource Manager; JSON: JavaScript Object Notation; XMLA: Extensible Markup Language for Analysis; C#: C-sharp; SaaS: Software as a Service; PaaS: platform as a Service

Acknowledgements

The author would like to thank the technical support from the department of IT in the Schlumberger Oil industry to gain the information and data gathering for program development.

Appendix 1

Prerequisites of Build Agent Hardware & Software:

- Windows 7, 8.1, or 10 (if using a client OS)
- Windows 2008 R2 SP1 or higher (if using a server OS)
- Power Shell 3.0 or higher
- .NET Framework 4.6.2 or higher
- For Building AAS Projects VS extension needs to be manually installed on the Visual Studio Interface https://marketplace.visualstudio.com/items?itemName=ProBI Tools.MicrosoftAnalysisServicesModelingProjects
- Open Visual Studio >> Go to Extensions Menu >> Click Manage Extensions >> Online >> Search for Microsoft Analysis Services Modeling Projects >> Install
- For deploying DACPACs, SqlPackage.exe is used and can be installed with Visual Studio 2019. Visual Studio 2019 installs the SqlPackage.exe at: C:\Program Files (x86)\Microsoft Visual Studio\2019\ {Your VS Version} \Common7\IDE\Extensions\Microsoft\SQLDB\DAC\150.
- For deploying SQL Scripts, Invoke-Sqlcmd cmdlet is used, and can be installed by the following -
- Download and install the following components of Microsoft SQL Server 2016 Feature Pack . On an x86 machine, only install the x86 version of the MSIs, and on an x64 machine, install both the x86 and the x64 versions of the MSIs.
- Microsoft® System CLR Types for Microsoft SQL Server® 2016 (SQLSysClrTypes.msi).
- Microsoft® SQL Server® 2016 Shared Management Objects (SharedManagementObjects.msi).
- Microsoft® Windows Power Shell Extensions for Microsoft SQL Server® 2016 (PowerShellTools.msi).
- Recommended:
- Visual Studio build tools (2015 or higher)
- Network:
- Agent Machine shall have access to the working directory set by/from Azure DevOps (Access to C: Drive for Agent Pool installation)
- Agent Machine shall have access to Azure DevOps
- Agent Machine shall have access to any Firewall within the environment
- Agent Machine shall have access to artifacts and infrastructure
- Note:
- Starting December 2019, the minimum required .NET version for build agents is 4.6.2 or higher.

Citation

1-Mojtaba Shahin; Muhammad Ali Babar Centre for Research on Engineering Software Technologies, The University of Adelaide, Adelaide, SA, Australia; Liming Zhu, "Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices ", INSPEC Accession Number: 16811548;DOI: 10.1109/ACCESS.2017.2685629;Publisher: IEEE; Date of Publication: 22 March 2017;Page(s): 3909 – 3943;Published in: IEEE Access (Volume: 5)

Ethical approval: This article does not contain any studies with human participants or animals performed by any of the authors.

REFERENCES

1. Schlumberger Azure Data warehouse: BDTDataPond: To consume data into Tableau/PowerBI for Power BI reporting: Business Owner (aka BPM, Product Owner, BPSM): Rodrigo Patrianova

2. Microsoft Power BI Desktop: Version: Version: 2.79.5768.721 64-bit (March 2020); PBIDesktopSetup_x64.exe;URL: https://www.microsoft.com/en-us/download/details.aspx?id=58494
3. Microsoft Visual Studio Community 2019: Version 16.5.5; URL: https://visualstudio.microsoft.com/downloads/
4. Microsoft Azure Portal: To build, manage, and monitor all the apps in Microsoft Azure Portal. A single, unified hub built for the teams and projects. :https://portal.azure.com
5. Schlumberger Enterprise data warehouse: glbbirptsqlsvrdev.database.windows.net
6. Schlumberger Azure Analysis Server: asazure: //westeurope.asazure.windows.net
7. Schlumberger application: Power BI pathway on Degreed : https://degreed.com/paths?path=getting-started-with-power-bi-enterprise-data-models&id=147477&orgsso=schlumberger
8. Schlumberger Azure Analysis Server: asazure://westeurope.asazure.windows.net
9. Microsoft SQL Server Management Studio 15.0.18206.0
10. Microsoft Analysis Services Client Tools 15.0.1567.0
11. Microsoft Data Access Components (MDAC) 10.0.17134.1
12. Microsoft MSXML 3.0 6.0
13. Microsoft Internet Explorer 9.11.17134.0
14. Microsoft .NET Framework 4.0.30319.42000
15. Operating System10.0.17134
16. Power Shell 3.0 https://docs.microsoft.com/powershell/scripting/install/installing-windows-powershell
17. .NET Framework https://docs.microsoft.com/dotnet/framework/install/
18. Microsoft SQL Server 2016 Feature Pack https://www.microsoft.com/en-s/download/details.aspx?id=52676
19. VS extension https://marketplace.visualstudio.com/ items?item Name=ProBITools.MicrosoftAnalysisServicesModelingProjects
20. SQL Scripts Deployment: https://msdn.microsoft.com/en-IN/library/cc281720.aspx

****