

# CA Release Automation Plugin Version 1.1.0

## Integration with Team Foundation Server 2015



Author	Manjunath Narayan
Team	Customer Advocate Team, Continuous Delivery
Version	2.0
File Name	Release Automation integration with TFS 2015
Date	12/05/2016

## Table of Contents

Introduction.....	3
Assumption.....	3
TFS Plugin Download .....	3
TFS Plugin Installation.....	4
Adding the Release Automation Deployment Step in the Build Definition .....	8
Option 1: Run Existing Deployment Plan .....	10
Option 2: Create new Deployment Plan .....	13

## Introduction

This documents will you guide to integrate Release Automation with Team Foundation Server 2015.  
We have used **CA Release Automation Plugin Version 1.1.0**

## Reference

Please find more information about the CA Release Automation Plug-In [HERE](#)

If you are using CA Release Automation Plugin Version 1.0.1, Please refer the document [HERE](#)

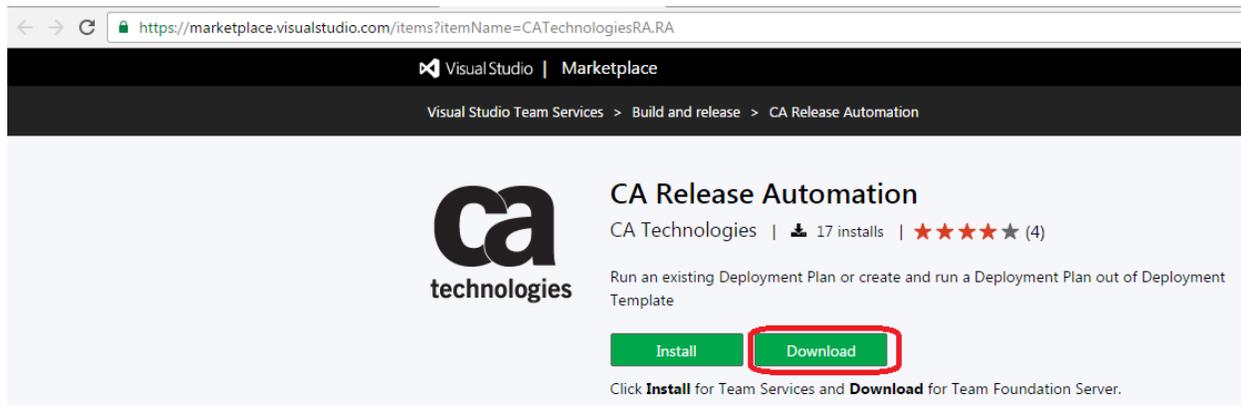
## Assumption

It is assumed that you have set up the below prerequisites

- 1) TFS 2015 Server (Update 2.1 and higher ) is Installed
- 2) All the required PORTs for TFS / VSTS is open
- 3) Build Agent is Installed and Configured
- 4) Build Definition is created
- 5) Required Software is already installed
- 6) Release Automation (Version 6.1 and above ) is installed
- 7) Processes are already created

## TFS Plugin Download

Download the CA- RA - TFS Plugin from Microsoft Market Place [Link](#)



Visual Studio | Marketplace

Visual Studio Team Services > Build and release > CA Release Automation

**CA Release Automation**  
CA Technologies | 17 installs | ★★★★★ (4)

Run an existing Deployment Plan or create and run a Deployment Plan out of Deployment Template

**Install** **Download**

Click **Install** for Team Services and **Download** for Team Foundation Server.

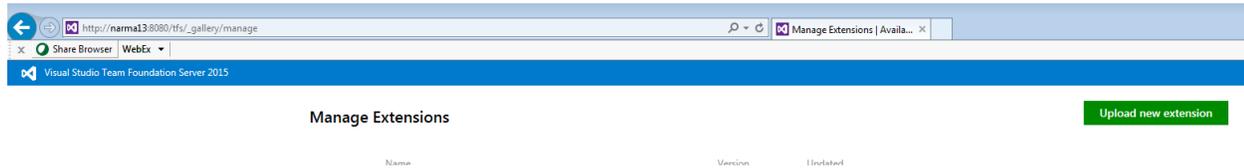
Download “**CATechnologiesRA.RA-1.1.0.vsix**” file

## TFS Plugin Installation

Go to server path

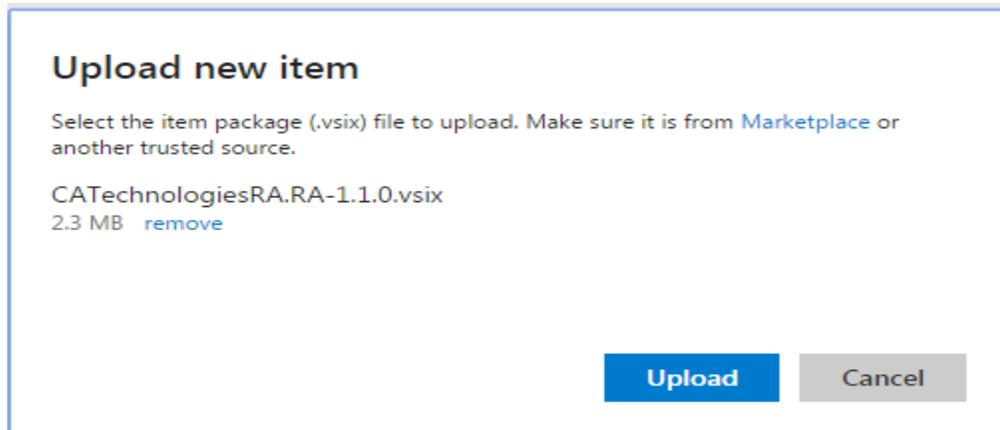
For example, [http://TFS-server:8080/tfs/\\_gallery/manage](http://TFS-server:8080/tfs/_gallery/manage)

Example:



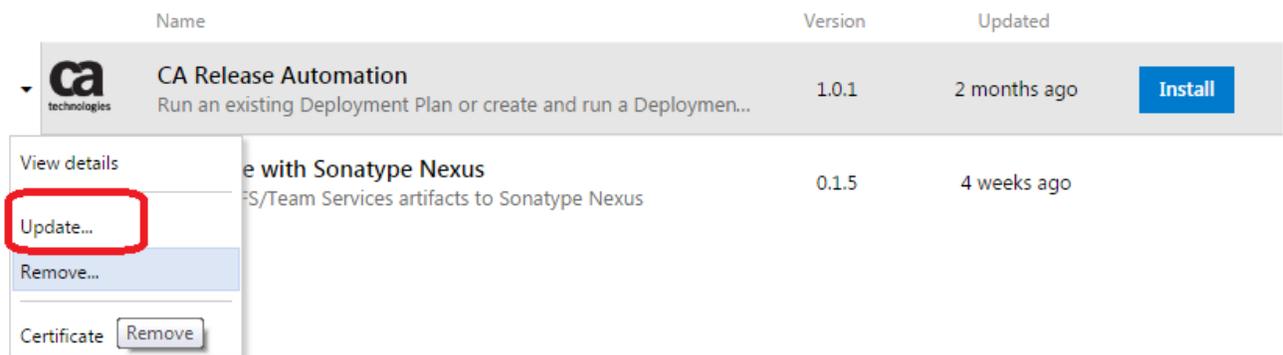
**Upload new extension**

Select **Upload new extension** and choose the **“CATechnologiesRA.RA-1.1.0.vsix”** file and select **“Upload”**



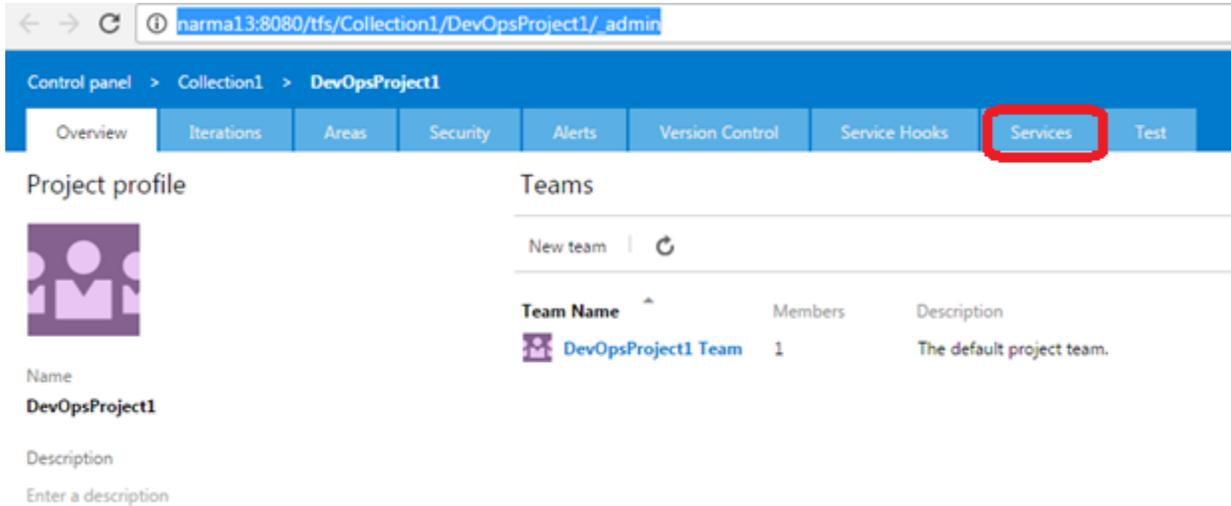
NOTE: If you have already uploaded the older version, you can update the older version  
Right Click on the Older Version and Select Update

### Manage Extensions

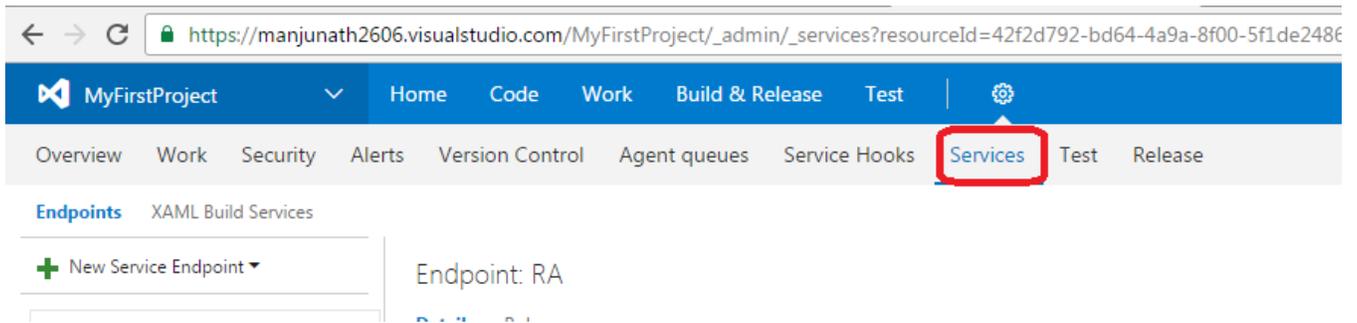


### Navigate to TFS SERVER admin page

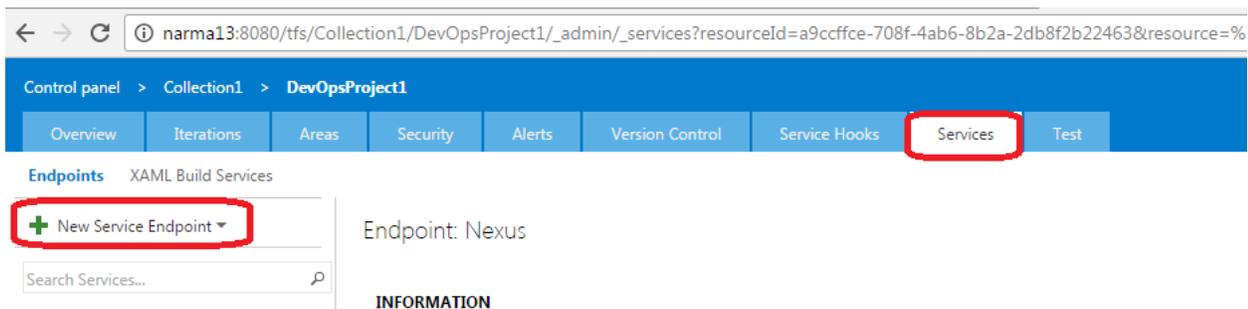
Example1: TFS SERVER [TFSSERVER:PORT/Collection/project/\_admin]



### Example2: VSTS Server



Go to the Services tab and select  to add CA Release Automation server details



Enter the below information

Connection Name: Desired Name for the connection [Example: RA]

Server URL: Release Automation Server URL

User Name: Release Automation Admin User

Password: Automation Admin Password

## Update Authentication for RA

Connection name

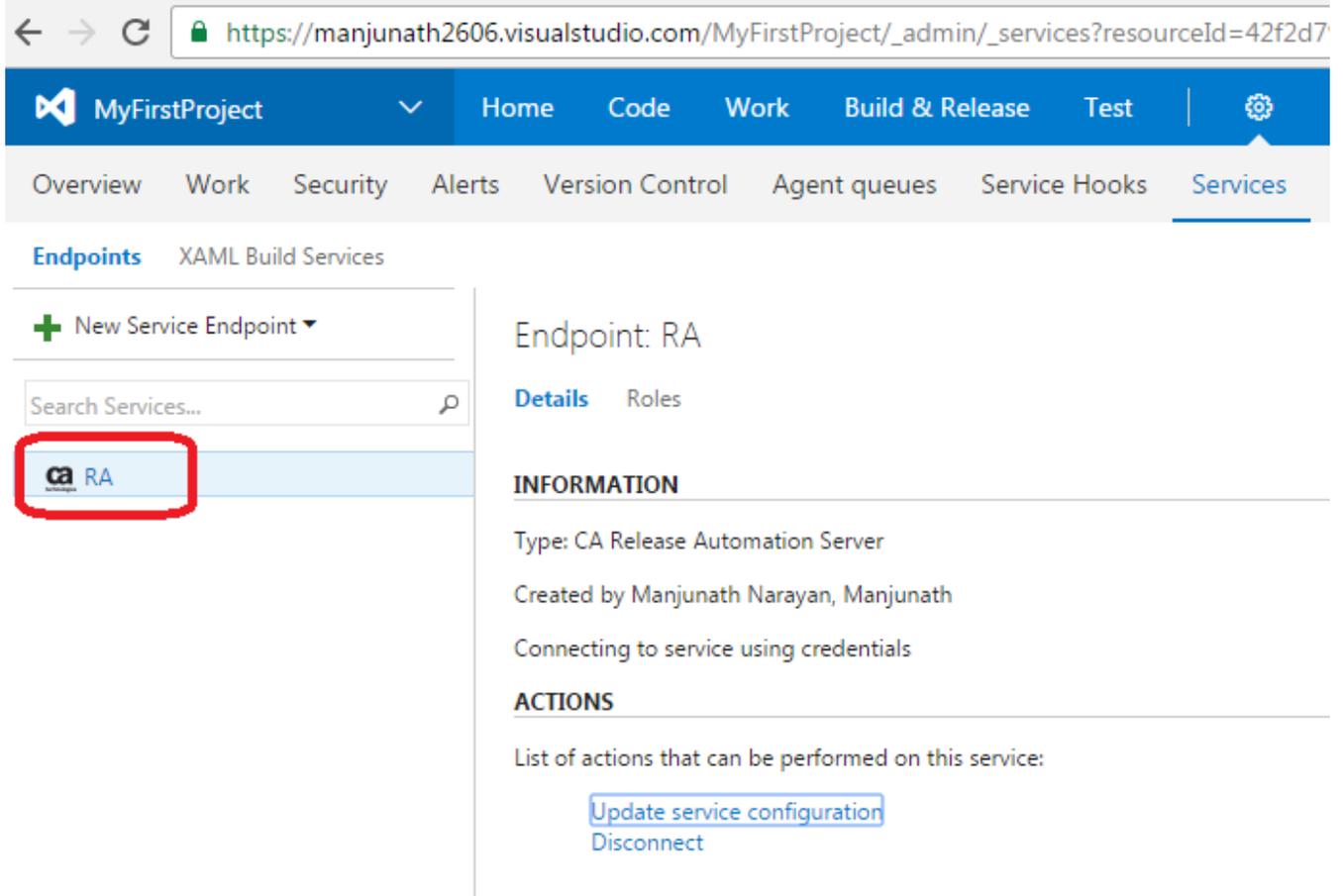
Server URL

Username  ⓘ

Password  ⓘ

Server URL i.e. <http://my-server:8080>, <https://my-secured-server:8443> etc...

Click on “**OK**” to complete the configuration



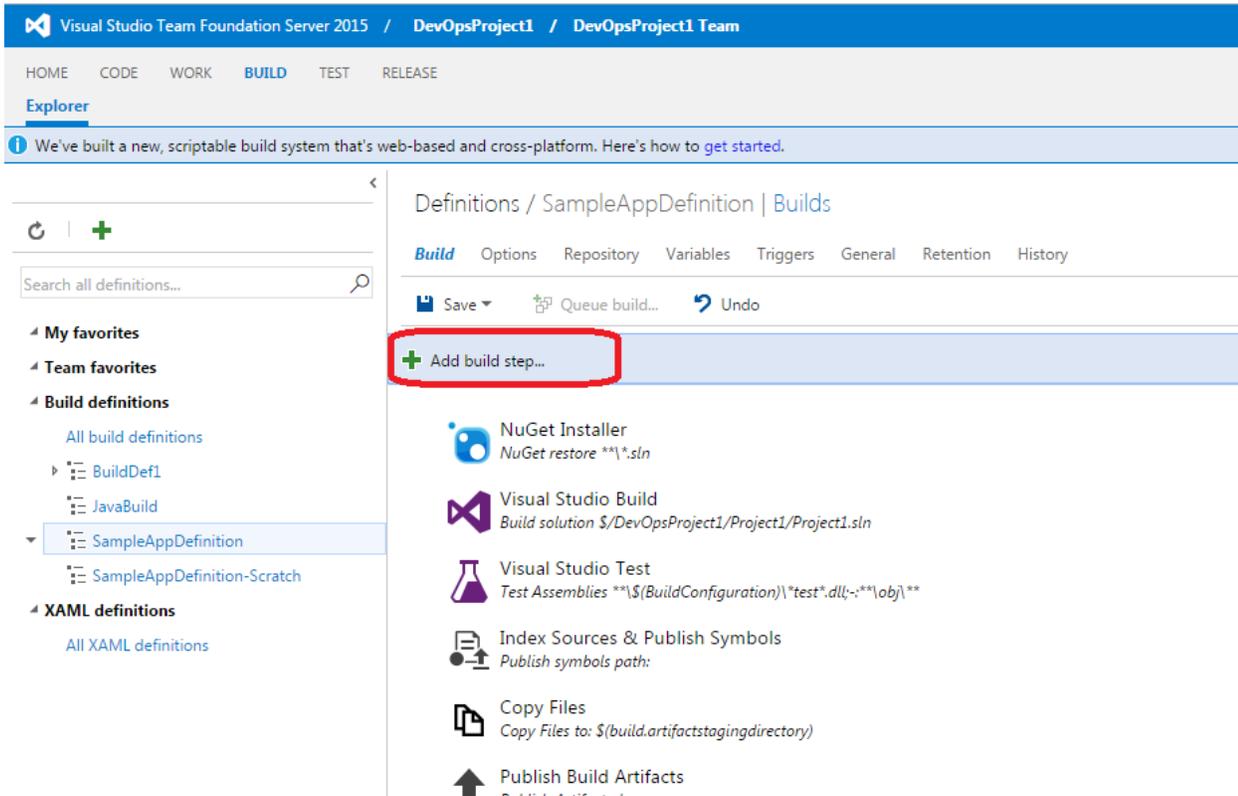
The screenshot shows the Visual Studio web interface. The browser address bar displays the URL: [https://manjunath2606.visualstudio.com/MyFirstProject/\\_admin/\\_services?resourceId=42f2d7](https://manjunath2606.visualstudio.com/MyFirstProject/_admin/_services?resourceId=42f2d7). The navigation bar includes 'MyFirstProject' and tabs for 'Home', 'Code', 'Work', 'Build & Release', 'Test', and 'Services'. Below the navigation bar, there are sub-tabs for 'Overview', 'Work', 'Security', 'Alerts', 'Version Control', 'Agent queues', 'Service Hooks', and 'Services'. The 'Services' sub-tab is active. On the left, under 'Endpoints', there is a search bar with the text 'Search Services...'. Below the search bar, a list of services is shown, with 'ca RA' highlighted and circled in red. The main content area displays the details for the 'Endpoint: RA'. It includes tabs for 'Details' and 'Roles'. Under the 'INFORMATION' section, it lists: 'Type: CA Release Automation Server', 'Created by Manjunath Narayan, Manjunath', and 'Connecting to service using credentials'. Under the 'ACTIONS' section, it lists: 'Update service configuration' and 'Disconnect'.

## Adding the Release Automation Deployment Step in the Build Definition

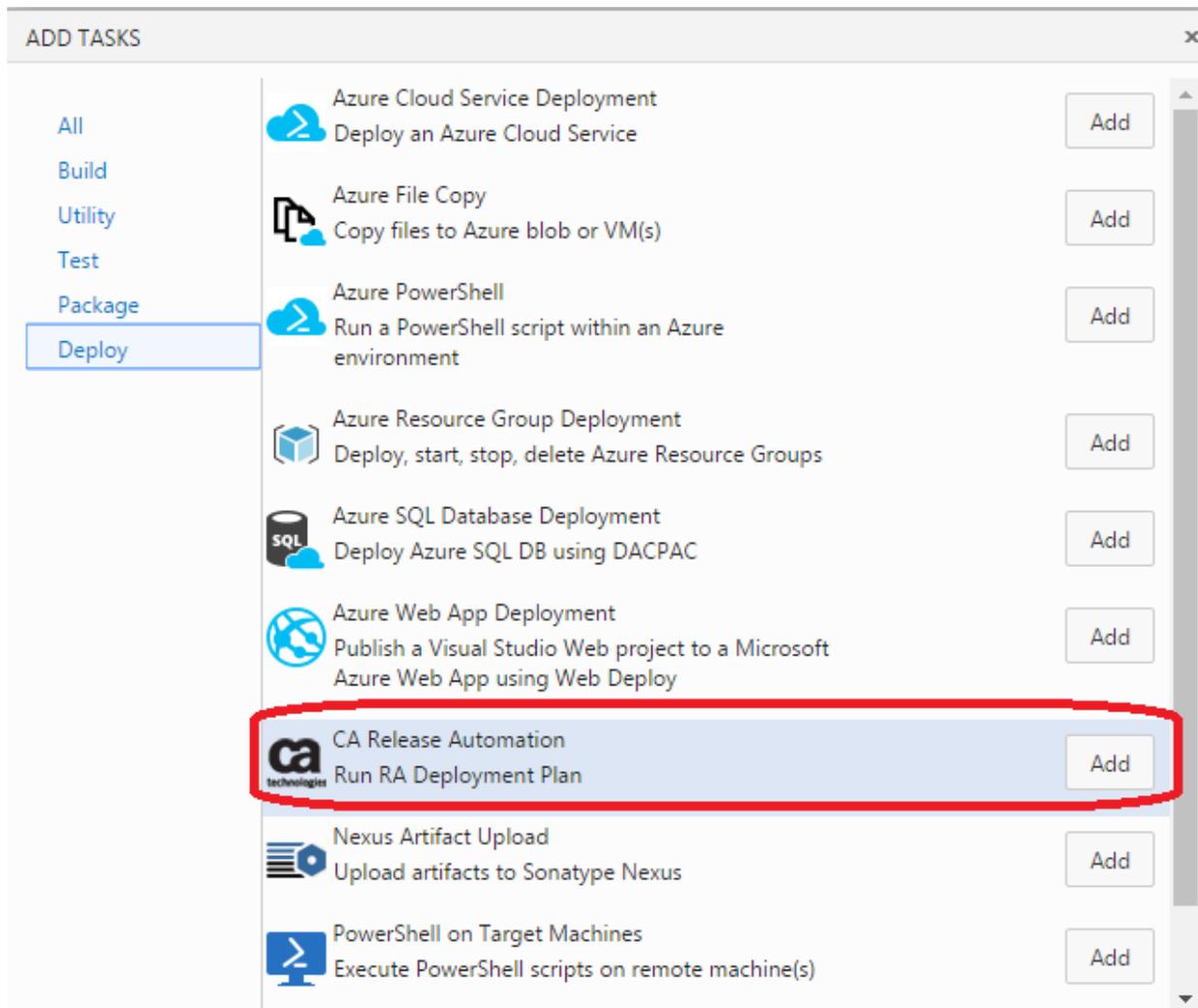
Navigate to Project Collection Page to Create the Build Definition of the project

Example: Editing the **SampleAppDefinition (Name of the Build Definition)**

Select **“Add build Step“**



GO to “Deploy” and Select “CA Release Automation” and Add



Category	Task Name	Description	Action
Deploy	Azure Cloud Service Deployment	Deploy an Azure Cloud Service	Add
	Azure File Copy	Copy files to Azure blob or VM(s)	Add
	Azure PowerShell	Run a PowerShell script within an Azure environment	Add
	Azure Resource Group Deployment	Deploy, start, stop, delete Azure Resource Groups	Add
	Azure SQL Database Deployment	Deploy Azure SQL DB using DACPAC	Add
	Azure Web App Deployment	Publish a Visual Studio Web project to a Microsoft Azure Web App using Web Deploy	Add
	<b>CA Release Automation</b>	<b>Run RA Deployment Plan</b>	<b>Add</b>
	Nexus Artifact Upload	Upload artifacts to Sonatype Nexus	Add
	PowerShell on Target Machines	Execute PowerShell scripts on remote machine(s)	Add

## Option 1: Run Existing Deployment Plan

### Enter the Below Details:

- RA Server: Choose the RA plugin connection from the dropdown.
- Strategy: Choose “Run existing Deployment Plan” from the dropdown.
- Application Name: Enter the existing Application from RA Server
- Project Name: Enter the existing Project which the deployment plan should relate to in RA Server
- Build Number: The Deployment plan build number as you defined when creating the deployment plan, not TFS build number.
- Deployment Name: The name of the new created deployment. You may use [TFS variables](#) in all of these fields
- Deployment Description: Enter the Deployment Description
- Deployment Stage: Enter Deployment Stage you wish to execute. All the preceding stages will be executed.
- Environment: The name of the existing environment to be deployed

**Note1:** You may use [TFS variables](#) in all of these fields

### Sample:

**Run Deployment Plan HelloWorldTFSPProject\_Deployment\_6** 

RA Server	RA <span style="float: right;"> <a href="#">Manage</a> </span>
Strategy	Run Existing Deployment Plan 
Application Name	SampleWebApplication 
Project Name	HelloWorldTFSPProject 
Deployment Plan Name	HelloWorldTFSPProject_Deployment_6 
Build number	6 
Deployment Name	Deployment_6 
Deployment Description	Deployment Description 
Deployment Stage To Perform	Deployment 
Environment	QA 

▶ **Advanced options for "Create new Deployment Plan" strategy**

▲ **Control Options**

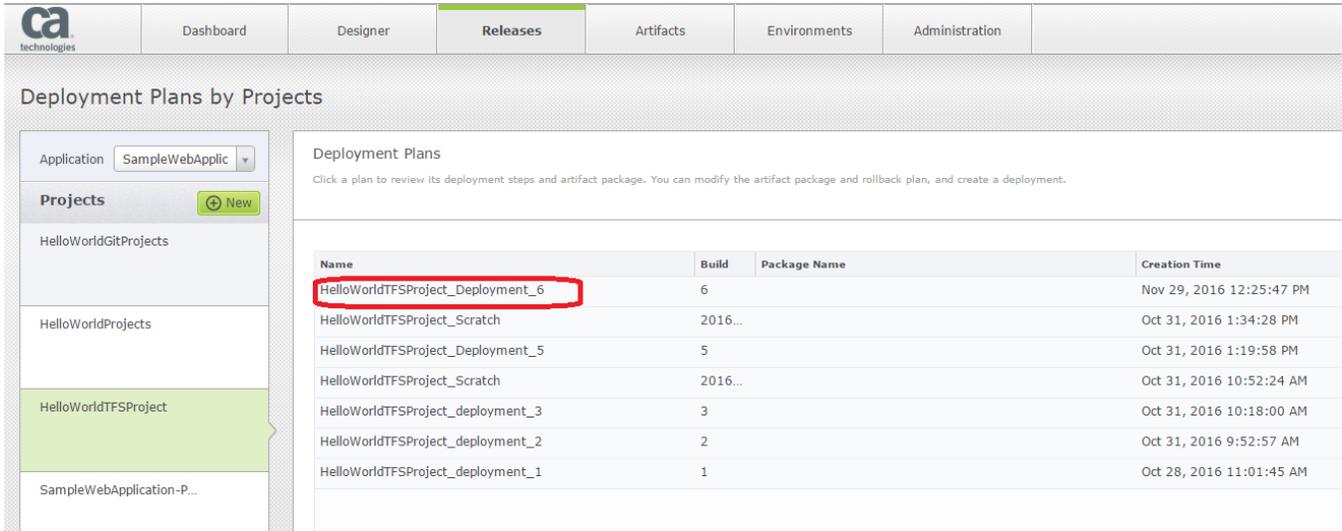
Enabled

Continue on error

Always run

**Note1:** Deployment Plan “**HelloWorldTFSPProject-Deployment\_6**” should already exist on Release Automation server as we are running an Existing Deployment Plan

Queue the Build in TFS and Deployment will be started on the Release Automation Server when it reaches the Deployment step in the Build Definition



Deployment Plans by Projects

Application: SampleWebApplic

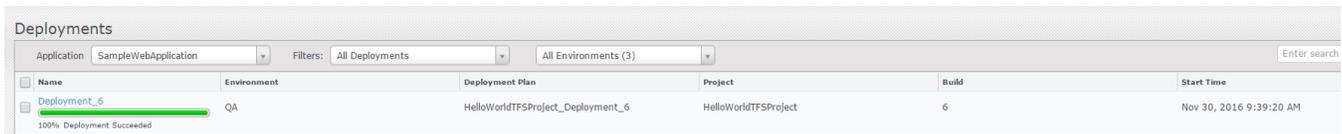
Projects: HelloWorldGitProjects, HelloWorldProjects, HelloWorldTFSPProject, SampleWebApplication-P...

Deployment Plans

Click a plan to review its deployment steps and artifact package. You can modify the artifact package and rollback plan, and create a deployment.

Name	Build	Package Name	Creation Time
HelloWorldTFSPProject_Deployment_6	6		Nov 29, 2016 12:25:47 PM
HelloWorldTFSPProject_Scratch	2016...		Oct 31, 2016 1:34:28 PM
HelloWorldTFSPProject_Deployment_5	5		Oct 31, 2016 1:19:58 PM
HelloWorldTFSPProject_Scratch	2016...		Oct 31, 2016 10:52:24 AM
HelloWorldTFSPProject_deployment_3	3		Oct 31, 2016 10:18:00 AM
HelloWorldTFSPProject_deployment_2	2		Oct 31, 2016 9:52:57 AM
HelloWorldTFSPProject_deployment_1	1		Oct 28, 2016 11:01:45 AM

You can notice the Deployment is started on Release Automation Server



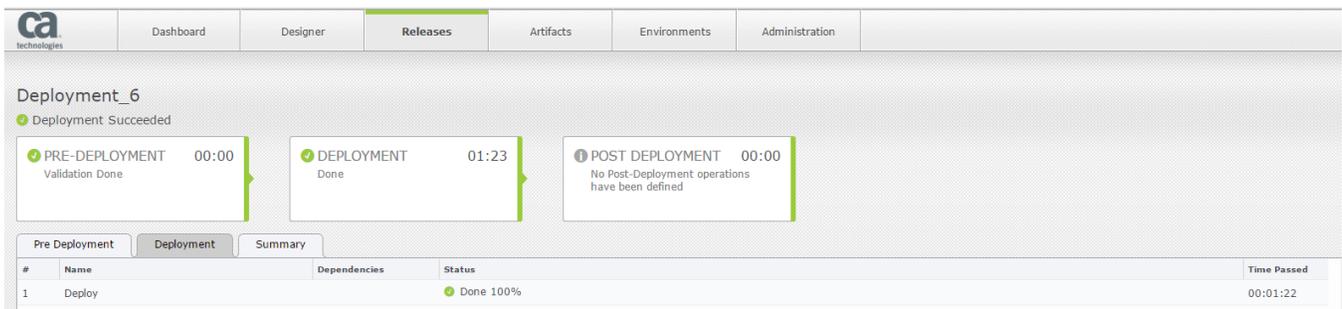
Deployments

Application: SampleWebApplication | Filters: All Deployments | All Environments (3)

Name	Environment	Deployment Plan	Project	Build	Start Time
Deployment_6	QA	HelloWorldTFSPProject_Deployment_6	HelloWorldTFSPProject	6	Nov 30, 2016 9:39:20 AM

100% Deployment Succeeded

Deployment Step is complete



Deployment\_6

Deployment Succeeded

- PRE-DEPLOYMENT 00:00: Validation Done
- DEPLOYMENT 01:23: Done
- POST DEPLOYMENT 00:00: No Post-Deployment operations have been defined

#	Name	Dependencies	Status	Time Passed
1	Deploy		Done 100%	00:01:22

## Editing the Build Definition after upgrading the Plug-in

Go to the Build Definition and go to the CA Release Automation Step and Select  Refresh button

**Run Deployment Plan HelloWorldTFSPProject\_Deployment\_6** 

RA Server	RA	 Manage 
Strategy	Run Existing Deployment Plan	
Application Name	SampleWebApplication	
Project Name	HelloWorldTFSPProject	
Deployment Plan Name	HelloWorldTFSPProject_Deployment_6	
Build number	6	
Deployment Name	Deployment_6	
Deployment Description		
Deployment Stage To Perform	Deployment	
Environment	QA	

▶ **Advanced options for "Create new Deployment Plan" strategy**

▲ **Control Options**

Enabled	<input checked="" type="checkbox"/>
Continue on error	<input checked="" type="checkbox"/>
Always run	<input checked="" type="checkbox"/>

Add the RA server details from the dropdown and Save the Build Definition

**Run Deployment Plan HelloWorldTFSPProject\_Deployment\_6** 

RA Server	RA	 Manage 
Strategy	RA	

## Option 2: Create new Deployment Plan

### Enter the Below Details:

RA Server:	Choose the RA plugin connection from the dropdown.
Strategy:	Choose “Run Deployment Plan from Scratch” from the dropdown.
Application Name:	Enter the existing Application from RA Server.
Project Name:	Enter the existing Project which the deployment plan should relate from RA Server
Template Category:	The Existing Template name you have created in RA Server.
Deployment Template:	The existing Deployment Template name you have created in RA Server.
Deployment Plan Name:	The name of the new created Deployment Template name you have created in Release Automation
Build Number:	Enter deployment plan build number. Use \$(Build.BuildNumber) for TFS build number.
Deployment Name:	Enter the name of the newly created deployment. Use deployment_\$(Build.BuildNumber).
Deployment Stage:	The deployment stage you wish to execute. All The preceding stages will be executed.
Environment:	Enter the existing RA Environment to be deployed.
Timeout:	Maximum time to wait for RA Server to finish the deployment, default is 0 - meaning no timeout.

**Note1:** You may use [TFS variables](#) in all of these fields

You may also add these optional advanced fields in Create new Deployment Plan mode:

Artifact Package:	Add an artifact package to the new Deployment Plan. You can use an existing package or an XML which will create a new one.
Manifest:	Add a XML Manifest to supply values for Release Automation Release Parameters.
Properties:	Add a pairs of key-value to supply values to Release Automation Template Properties.

### Run Deployment Plan DeploymentPlan-\${JOB\_NAME}-\${build\_id}

RA Server	RA
Strategy	Create new Deployment Plan
Application Name	SampleWebApplication
Project Name	SampleWebApplication-Projects
Template Category	SampleWebApplication-TEMPLATE
Deployment Template	DEPLOYMENT-TEMPLATE
Deployment Plan Name	DeploymentPlan-\${JOB_NAME}-\${build_id}
Deployment Plan Description	dDESCRIPTION
Build number	\$(Build.BuildNumber)
Deployment Name	Deployment_\$(Build.BuildNumber)
Deployment Description	Deployment Description
Deployment Stage To Perform	Deployment
Environment	QA
Timeout	200

#### Advanced options for "Create new Deployment Plan" strategy

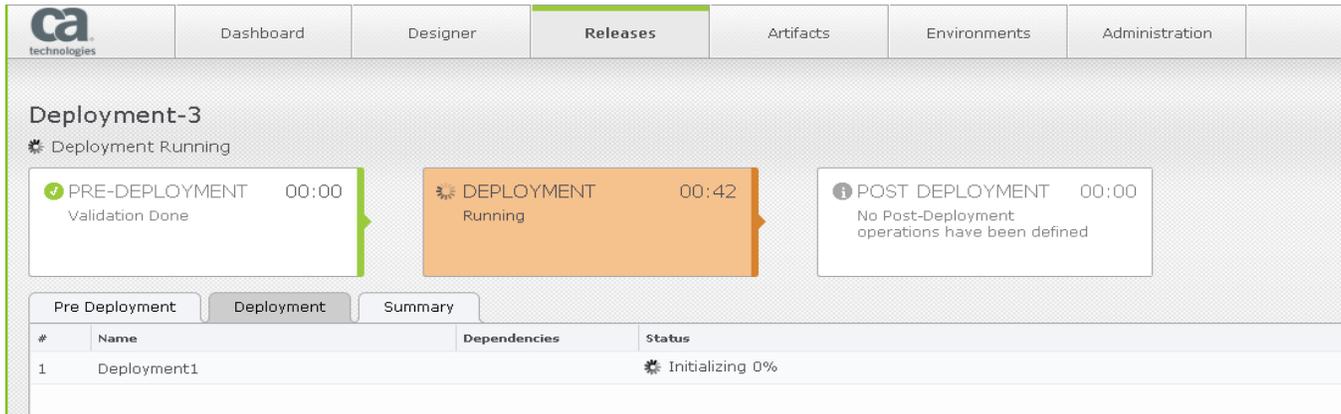
Artifacts Package as...	<input type="radio"/> Package Name <input checked="" type="radio"/> XML
Artifact Package as XML	<input type="text"/>
Manifest	<input type="text"/>
Properties	<input type="text"/>

#### Control Options

Enabled	<input checked="" type="checkbox"/>
Continue on error	<input type="checkbox"/>
Always run	<input type="checkbox"/>
Timeout	<input type="text" value="0"/>

 CA 

When you run the build, when it reaches the Deployment Step, deployment will start



**Deployment-3**  
 \* Deployment Running

✓ PRE-DEPLOYMENT 00:00  
 Validation Done

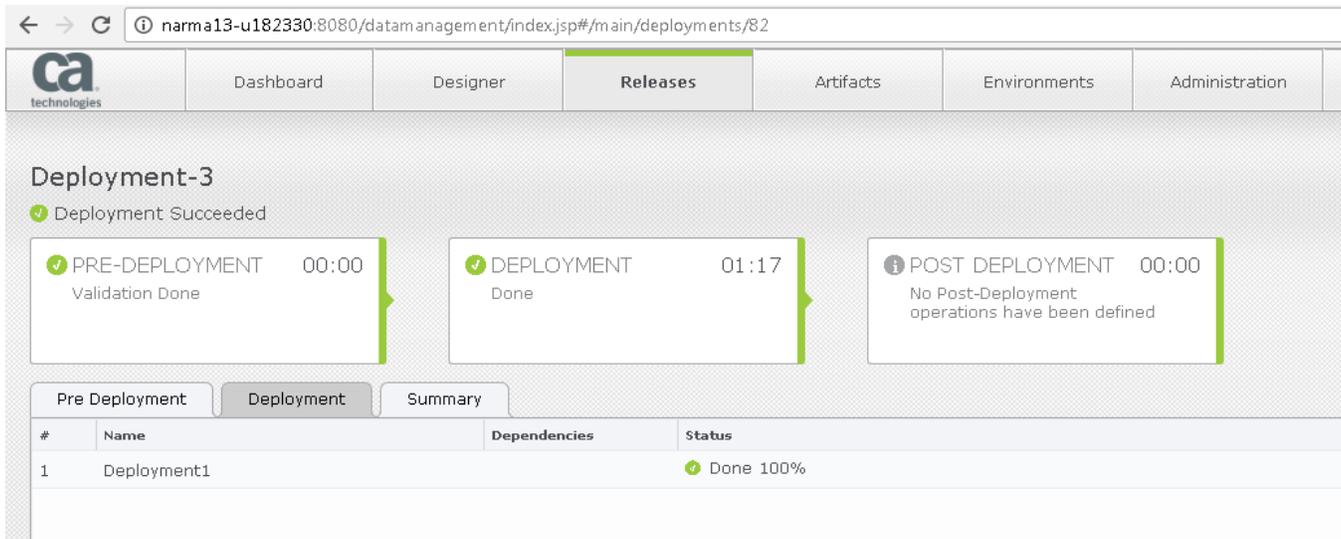
\* DEPLOYMENT 00:42  
 Running

i POST DEPLOYMENT 00:00  
 No Post-Deployment operations have been defined

Pre Deployment | **Deployment** | Summary

#	Name	Dependencies	Status
1	Deployment1		* Initializing 0%

Deployment is completed



← → ↻ [nama13-u182330:8080/datamanagement/index.jsp#/main/deployments/82](#)

**Deployment-3**  
 ✓ Deployment Succeeded

✓ PRE-DEPLOYMENT 00:00  
 Validation Done

✓ DEPLOYMENT 01:17  
 Done

i POST DEPLOYMENT 00:00  
 No Post-Deployment operations have been defined

Pre Deployment | **Deployment** | Summary

#	Name	Dependencies	Status
1	Deployment1		✓ Done 100%