Icon

Description automatically generated



FIS – DevOps

Configuration

Revision History

|  |  |  |
| --- | --- | --- |
| Version | Author(s) | Summary of changes |
| 1 | MOHD GHOUSE | Introduction |
| 2 | MOHD GHOUSE | UAT PIPELINE |
| 3 | MOHD GHOUSE | Agent Installation |
| 4 | MOHD GHOUSE | Build Defiantion |
| 5 | MOHD GHOUSE | Alerts |
| 6 | MOHD GHOUSE | Release Defination |

**Table of Contents**

1 Introduction…………………………………………………………………………………………..3

2 UAT PIPELINE………………………………………………………………………………………….3

3. Agent Installation………………………………………………………………………………… 3

4. Build Defiantion………………………………………………………………………………………8

5. Alerts…………………………………………………………………………………………………….13

5. Release Defination…………………………………………………………………………………. 15

6. Approval and deployment condition……………………………………………………….19

# INTRODUCTION

This document has the intent to describe the DevOps processes performed at the FIS environment, and provide an explanation about the environments involved throught the process and the technologies utilized.

# Environment

Currently, working on one environment : UAT and for a better understand the diagram flows.

|  |
| --- |
| Uat |

The UAT environment is utilized to perform tests and validate the deployed code before going to PROD.

## 

## Agent installation

* Install the agent pool for UAT environment in [LRK1WFISTFSAP02] which also called the build server.
* An agent can be defined as an installable software that runs one build or deployment job at a time.

**Steps for configuring agent pool:**

* To configure the agent in an agent pool you should have administrator rights.
* Enter into the TFS URL [http://corpsystfs.fnfis.com:8080/tfs/\_admin/\_AgentPool?hubGroupId=ms.vss-web.account-admin-hub-group]
* This will open the agent pool for the project collection as shown in below image.

Graphical user interface, text, application

Description automatically generated

**Download the agent**

Click on the download agent link on the left side to download the zip file.

A picture containing diagram

Description automatically generated

It will ask you to save/open the zip file. Save the agent.zip file.

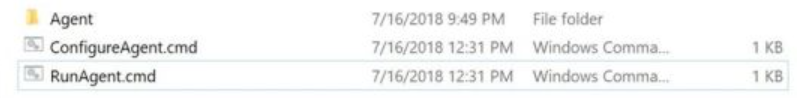


Extract the zip file.

Graphical user interface, application, table

Description automatically generated

This will give below content inside the Agent folder.



**Configure the Agent**

* To configure the agent right click on RunAgent.cmd and run as an administrator as shown.
* Now, right click on the ConfigureAgent and run as administrator.

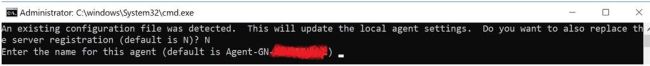
Graphical user interface, application

Description automatically generated

**Enter Details**  
cmd.exe opened in administrator mode with prompts.

* t will prompt about the replacement of server registration and update the local agent settings. Press N

1. *Agent Name*Enter Name for this agent [if you would like to change]. By Default, it’s taking your system name as a prefix by the word Agent as your agent name. Press enter for default one.



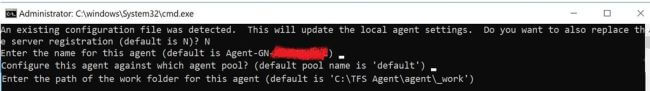
1. *TFS Server Name*Now, enter your TFS server Name and press enter. For ex: http://corpsystfs.fnfis.com:8080/tfs

Server

1. *App pool Name*Configure the agent under ‘Default’ Agent Pool and hit enter. [I'll explain another article for the detailed descriptions of Agent pools, Agent Queues, and agents. ]



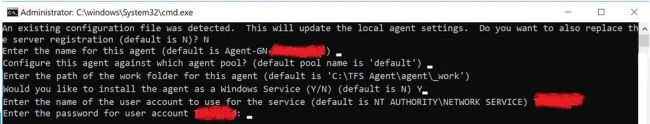
1. *Agent’s working folder*By Default, it will create a Folder \_work in the same location where you are configuring your agent. If you would like to make any other folder as Agent’s working directory, then provide the path here. Later this working folder is used to download the source code and run the builds. I selected the Default one and hit enter.



1. *Run Agent as Service*Now this is an important point here, I want to run the agent as a Windows Service, so press ‘Y’ and hit enter.



1. *Enter Administrative credentials for agent service*Provide credentials for the service. These credentials must have administrator rights.



1. Agent as a service installed successfully.

We can see the \_work folder under Agent as shown below.But in our case work folder is \_cpUpgradework

***Add capabilities to run build on particular agent.***

Graphical user interface, text, application, email

Description automatically generated

# **Creating a Build Definition in TFS 2017**

To run the Build we need t o create the build definition.

**Steps:**

* Once enter into the project seems a enter to build tab then seems a + icon to create the build definition.

Graphical user interface

Description automatically generated

* TO create build definition from scratch enter go with empty

Graphical user interface, text, application

Description automatically generated

Once click on Next seem to select the repository.Select the repository and click create.

Graphical user interface, application

Description automatically generated

**First task to nuget installer.**

The NuGet Installer step will be used to restore the NuGet packages before building the solution. The Visual Studio Build step will build all solutions that it will find in the configured path.

Prerequites for nuget installer.

4.8 Framework should must be install in build server [**LRK1WFISTFSAP02**].

List of below website should be accessible or **whitelist** from internet.

* <https://api.nuget.org/v3/index.json>
* <https://www.nuget.org/api/v2>
* <https://sitecore.myget.org/F/sc-packages/api/v3/index.json>

**Second Task to add msbuild.**

Graphical user interface, text, application, email

Description automatically generated

* In the project column give the sln path
* In this project using build platform should be Any CPU.
* BuildConfiguration is using based on pipeline in our project creating two different pipeline with two different configuration UATNewCM and UATNewCD1
* In third column MSBuild Arguments given as per below

[/p:DeployOnBuild=true /p:WebPublishMethod=FileSystem /p:SkipInvalidConfigurations=true /p:PackageLocation="$(build.artifactstagingdirectory)" /p:PublishUrl=$(build.artifactstagingdirectory) /p:DeployDefaultTarget=WebPublish]

**Third task to publish the artifact.**

Graphical user interface

Description automatically generated

* Path to Publish where want to publish in our case using $(build.artifactstagingdirectory)
* In second column give the artifact name.
* In this colum give the artifact type.

In **General** tab add the demands for run the build on creating agent pool

Graphical user interface, text, application

Description automatically generated

In this case we have add the capablies in agent pool where key is Agent.Name and value is CPUpgrade-BuildAgent.

**Now save the Build definition**

Graphical user interface, application

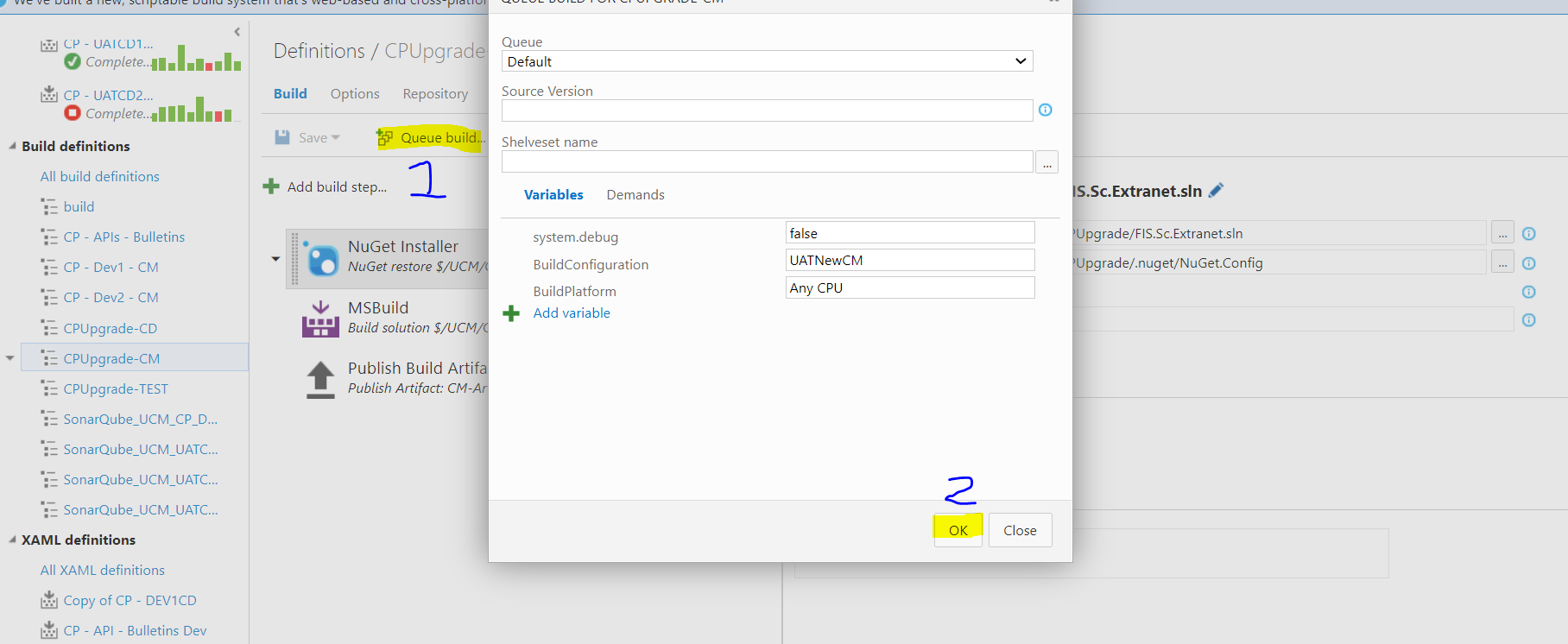
Description automatically generated

Once click on build rename the build definition if want and click ok.

Graphical user interface, application, Teams

Description automatically generated

To run the build definition click on queue the build.



**ALERTS:**

TFS Global Alerts. A web service to notify any number of users when build fails. Notification logic is easily customisable to suit your environment. Project Description. A web service to notify any number of users when builds gets failed

* Steps for configure the alerts in TFS
* Go to the TFS portal and select the project.
* Now click on settings,then control panel.
* In the third options seems the alerts. Click on the alerts.
* Create new alert and select the template when build fails
* In the send option add the email
* Separate multiple email addresses with a semi-colon (;) or comma (,)
* Alerts can also create if want at build successful.just need to change the template.

Background pattern

Description automatically generated with medium confidence

A picture containing bar chart

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

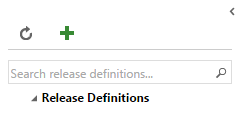
Graphical user interface, text, application

Description automatically generated

**Release Management**

By clicking the RELEASE menu item from the top menu of your **TFS** dashboard you open the Release Explorer dashboard. It is from here where you will create your release and deployment for the currently selected project. You will be presented with a list of any currently configured **Release Definitions** as well as the releases created from those **Release Definitions**.

To begin with click on the plus (+) sign above the **Release Definitions**



You will be presented with a list of deployment templates to choose from. For the purposes of this article I will select the **Empty** template, but feel free to select any of the other templates as necessary.

Click the **OK** button to proceed. You will now be presented with an empty **Release Definition** and associated Environment as in the screenshot below.

1. Give your **Release Definition** a meaningful name in the textbox provided e.g. *UAT-CM Release*
2. Configure your environment
   * Give the environment a meaningful name by clicking into the **Default Environment** name provided and enter one of your own e.g. UAT-CM

There is a second tab called **ARTIFACT**. This is used to link the build definition which is want to deploy.

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

In the third tab **configuration**.It is here where can add variable by clicking add variable .here can add user name,password or other secrets which can used in task.

Graphical user interface, text, application, email

Description automatically generated

In the fourth tab **trigger**. It is here that we define what triggers the deployment of our release.

* **Manual**- deployment of the build artefacts will need to be undertaken manually with no automation
* **Continuous Deployment**- Creates a new release every time a new artifact version is available.i.e. **Continuous Delivery**.
* **Schedule** - Create a new release at a specified time.

Graphical user interface, text, application, email

Description automatically generated

In the fifth **General** tab.Here can add the Release name fortmat and update the retention policy.

Graphical user interface, text, application, email

Description automatically generated

SIxth tab **History** is used to look the history of created build definition ex UAT-CM.

Graphical user interface, text, application

Description automatically generated

Click on the **Add tasks** button to create our deployment task i.e. the action that we want to happen when we invoke a deployment.

Graphical user interface, text, application

Description automatically generated

Select **Windows Machine File Copy**, click the **Add** button then click **Close**. Enter the necessary details as appropriate. An example screenshot is given below.

Graphical user interface, text, application, email

Description automatically generated

Make sure on deployment folder’s would be shared.

**Approvals:**

Select the three dots in environment where can seen the assign approval click it

Graphical user interface, text, application

Description automatically generated

You can then set Pre-deployment and Post-Deployment approvers.

In the Options section, select the **Send an email notification to the approver** option

Graphical user interface, text, application, email

Description automatically generated

Second tab **queue** is used to add demands.

Graphical user interface, text, application, email

Description automatically generated

Fourth tab **General** is used to add email notification and other general setting related to release.

Graphical user interface, text, application

Description automatically generated

Fifth tab **Deployment Conditions** is used to Define the trigger that will start deployment to this environment.

Graphical user interface, text, application, email

Description automatically generated