# **Azure portal -pre-requisites and Configuration:**

Sign in to <https://portal.azure.com> with valid credentials and subscription.

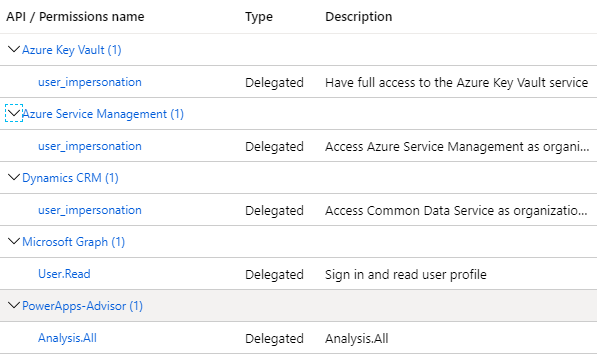
## **Azure Active Directory:**

Go to Azure active directory and register new application:

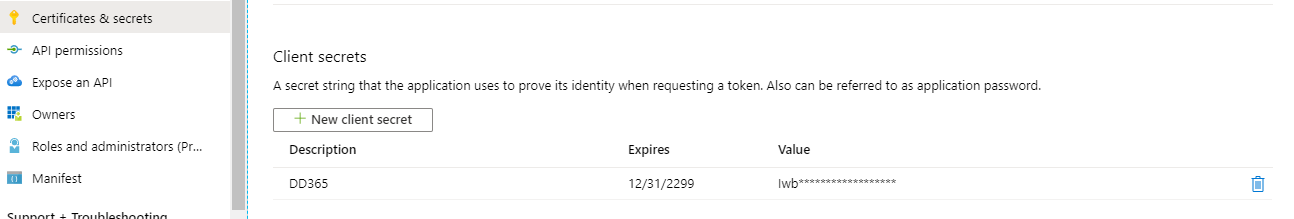
After Creating new application, Copy **Application Id** and **Tenant Id** to clipboard.

Give API permission to application, Click on API Permission:

* + - * Click on +Add permission
      * In Microsoft ApI’s Add Azure Key vault, Dynamic CRM APIs.
      * Go to APIs my organization uses
      * Search for PowerApps-Advisor Add permission to it.



Now go to certificate & Secrets section and create a new **Client Secret** and copy it to clipboard:



## **Create Azure Function:**

Download function app code Zip file from ..\..\Release\AzureFunction path to local drive.

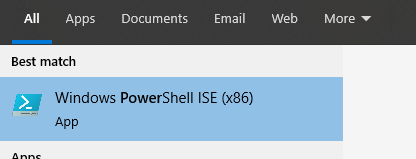


Create Azure storage Account, App service plan to deploy azure function in the portal.

Go to Function app in the portal and click on +Add button.

Select valid subscription, Resource Group, Function App Name, Region and click on **Review+Create** button.

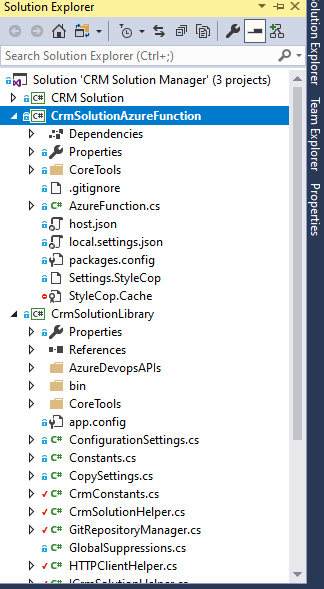
Now open PowerShell tool and run below CLI command with details :



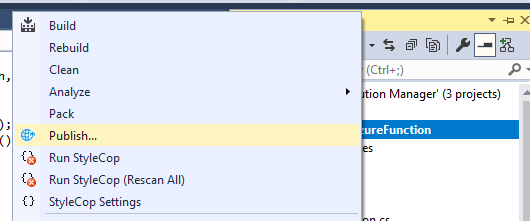
**az login  
az account set --subscription "subscriptonsname"  
az functionapp deployment source config-zip -g "resourcegrupo" -n "dd365compassazurefunciton" --src "E:\AllRepos\DynamicsPOC\CRM Solution Manager\Release\AzureFunction\CICDAzureFunction.zip"**

**Another way to deploy above azure function:** Download the above solution file and open it with visual studio and compile it .

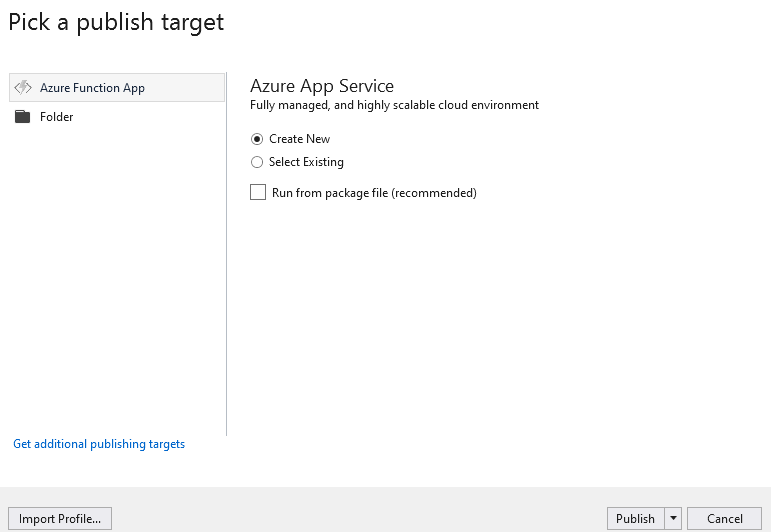
After compiling the solution successfully, right click on azure function and click on publish it to azure:



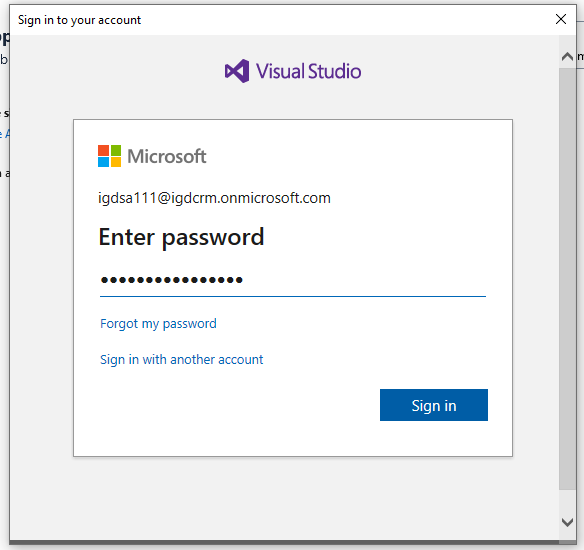
Right click on function:



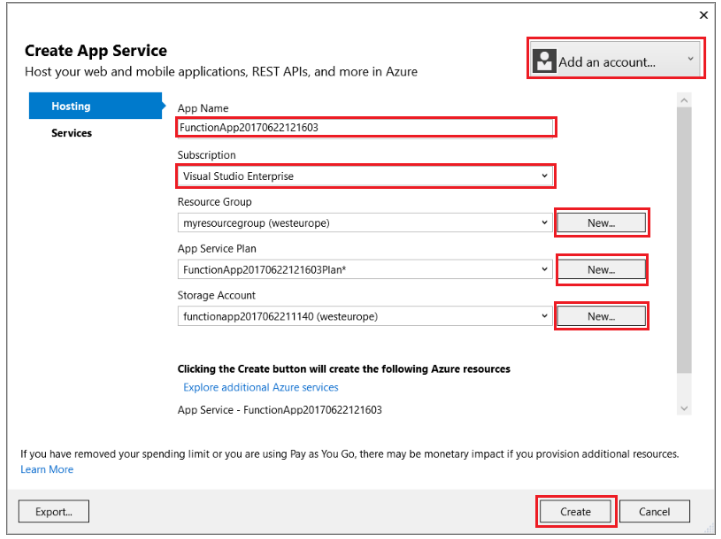
Click on publish:



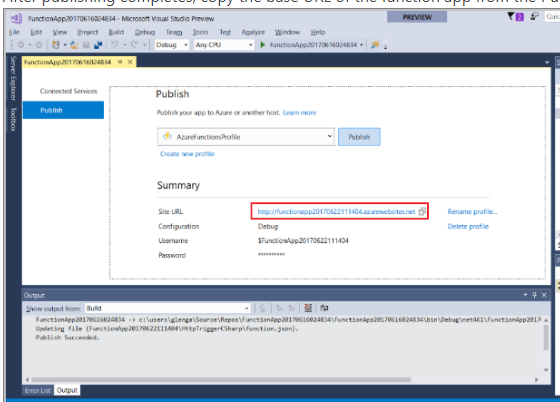
Click on publish and it will prompt for Sign in to azure :



After login create a app service to host azure function:



Then it will give function url and details :

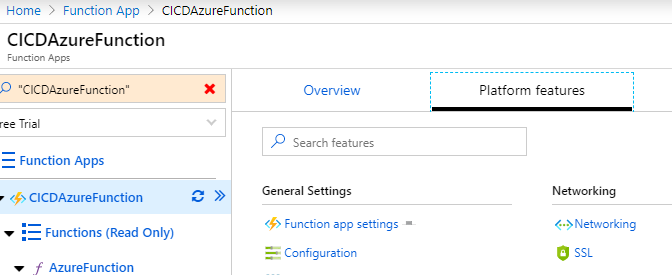


The URL should look like

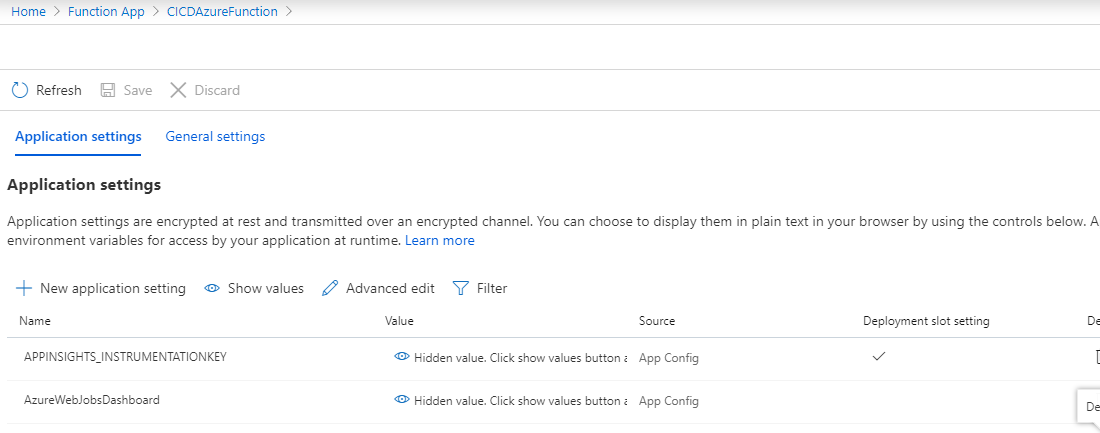
http://<functionappname>.azurewebsites.net/api/<functionname>?name=<yourname>

After running successfully, go to Azure portal and Create app key in the Azure function.

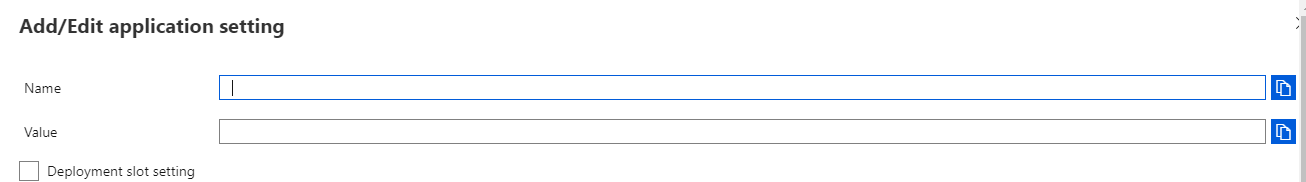
In the Function App, Go to Platform features:



Click on Configuration link:



Under application settings, click on +New Application setting:



Fallow the below details and Add Name and value. ("Name": "value")

"CRMSourceInstanceUrl": "",

"CRMSourceServiceUrl": "",

"CRMSourceUserName": "",

"CRMSourcePassword": "",

"GitUserName": "",

"GitPassword": "",

"TFSUser": "",

"TFSPassword": "",

"ClientApplicationSecret": "your application secret",

"SolutionCheckerAppClientId": "your application Id",

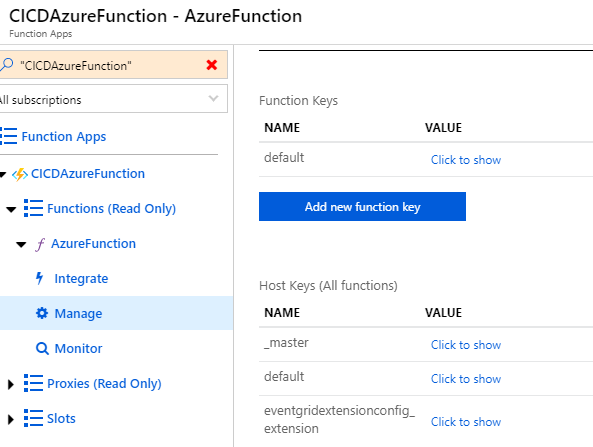
"TenantId": "your tenant Id",

"BASESECRETURI": <https://VaultName.vault.azure.net/> **(After creating Key vault we need to updated this).**

Now go back to function copy Azure function URL to clipboard



and click on manage link:



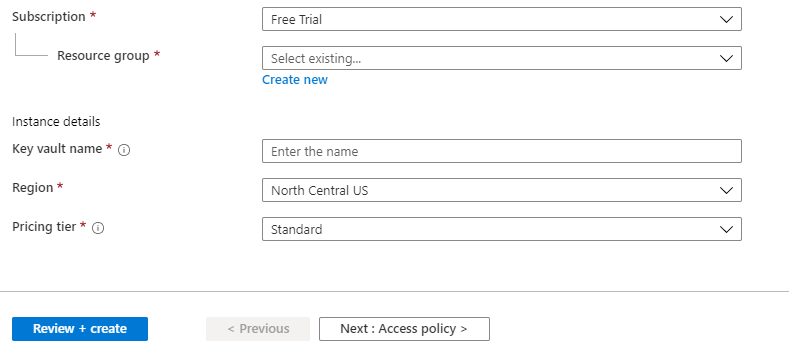
Click on \_Master **Click to show** link in the above screen:



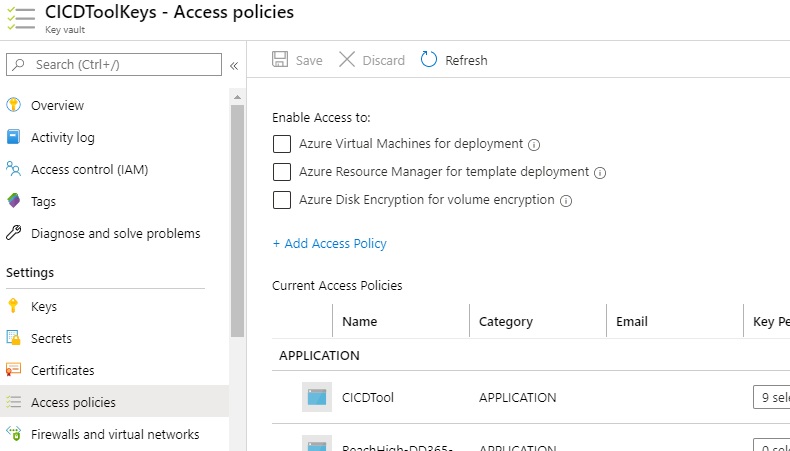
Click on copy link and copy master key to click board.

## **Azure Key vault creation:**

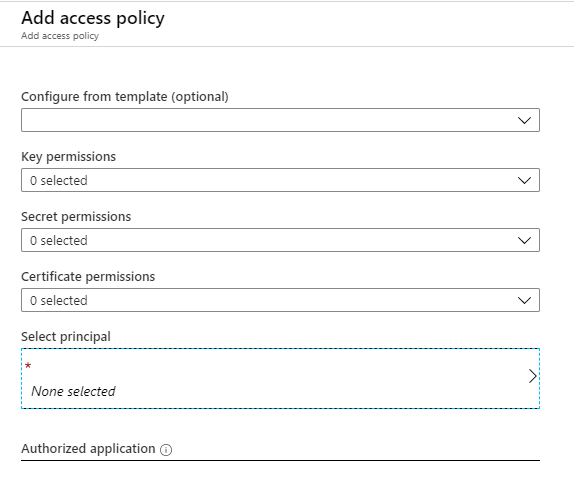
Go to Key vault and click on Add, give subscription,Resource group, key vault name, Region, pricing tier and click on **Review+Create button .**



After creating, go to key vault and click on access policies link:



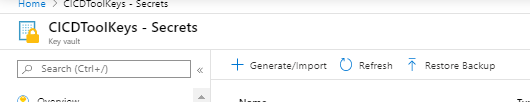
Now click on + Add Access Policy:



In the service principal search for Azure application name and select it and click on Add.



Now go to Secrets in the key vaults:



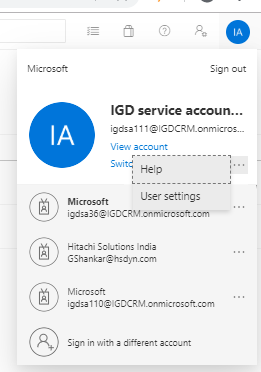
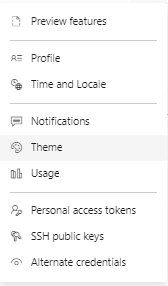
Click on +Generate/Import button and add the key secrets, please fallow the below excel to add the secrets .



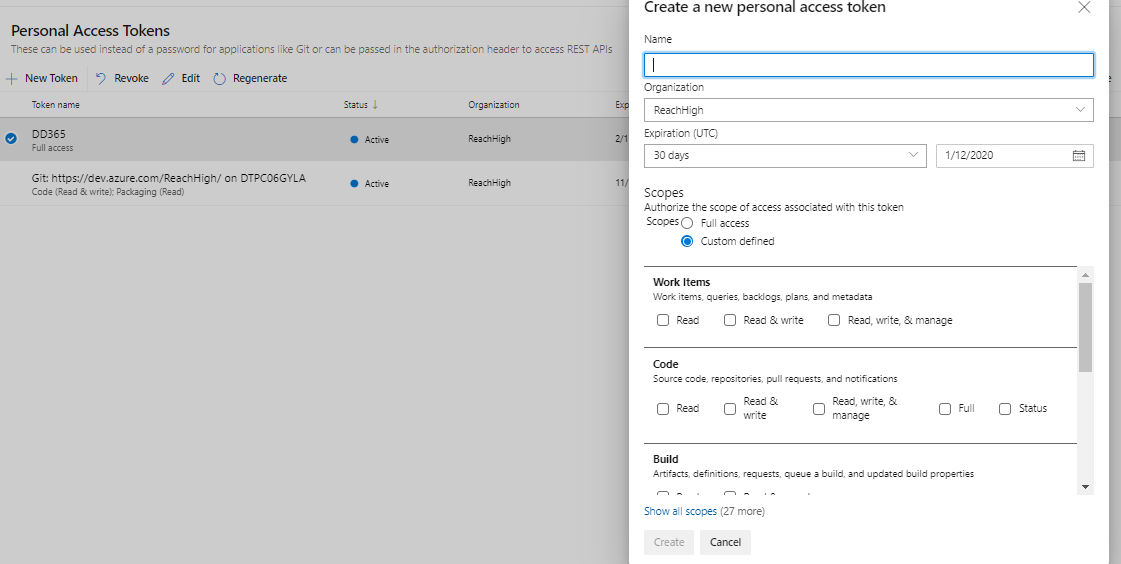
# **Devops Configuration:**

## **Personnel Access tokens**:

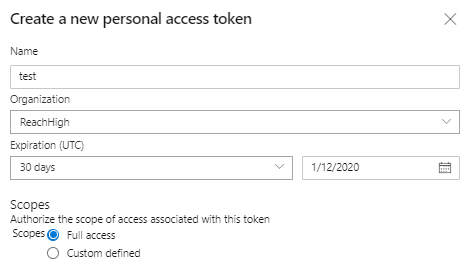
Login to devops (<https://dev.azure.com/>) and Create a PAT token, in the DevOps go to extreme right, select your user , click on …, click on User Settings, click Personnel Access Tokens, Generate New Token.

Create a new Access token:

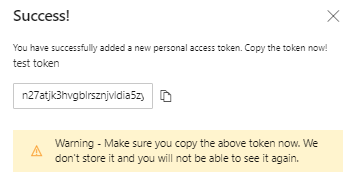


With Full Access:





Copy below created Access token: (need to update the GitPassword Key vault secret with this value)



Next Create a DD365 branch and upload below DD365 into the tools folder or any required folder in the branch.



In DD365 folder there are checkin and release folder which will be used by the tool to trigger build and release. These folders will be modified by the tool. Commit the branch and push the branch.

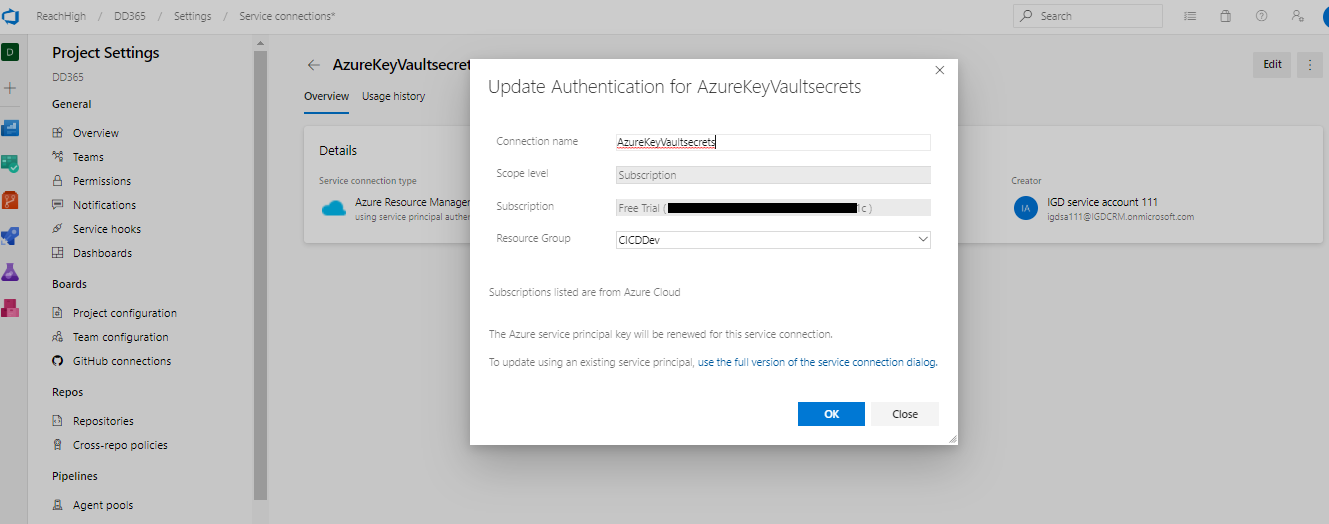
Configure Azure key Vault with the details as mentioned in the excel



## **Service Connection creation:**

Create Service connection to access Azure Key vaults secret values

* + Go to project settings
  + Click on service connections
  + New service connection
  + Select **azure resource manager** connection type template
  + Give the details and create it.



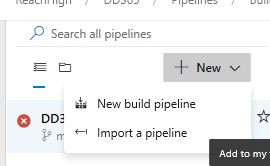
## **Configure Build Pipeline:**

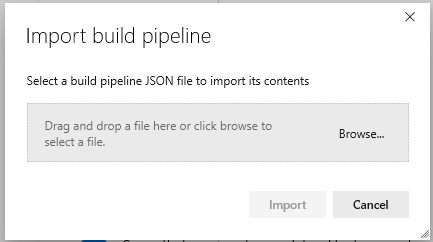
Open folder DynamicsPOC\CRM Solution Manager\Release\Build and Release template



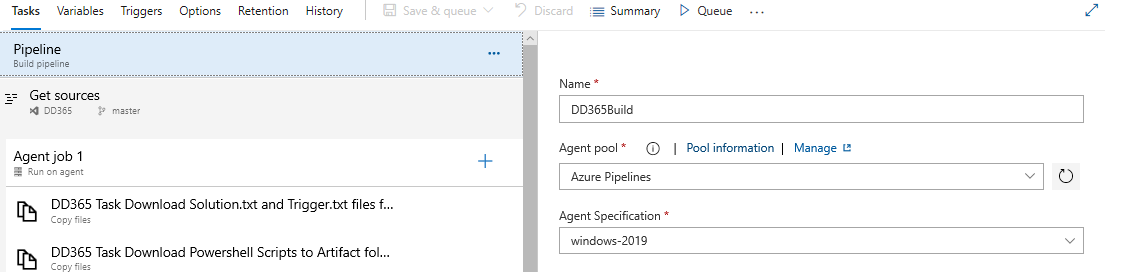
Go to Pipeline and build pipeline in devops:

* + Click on build, click on new, click on import a pipeline and select the build json from above path

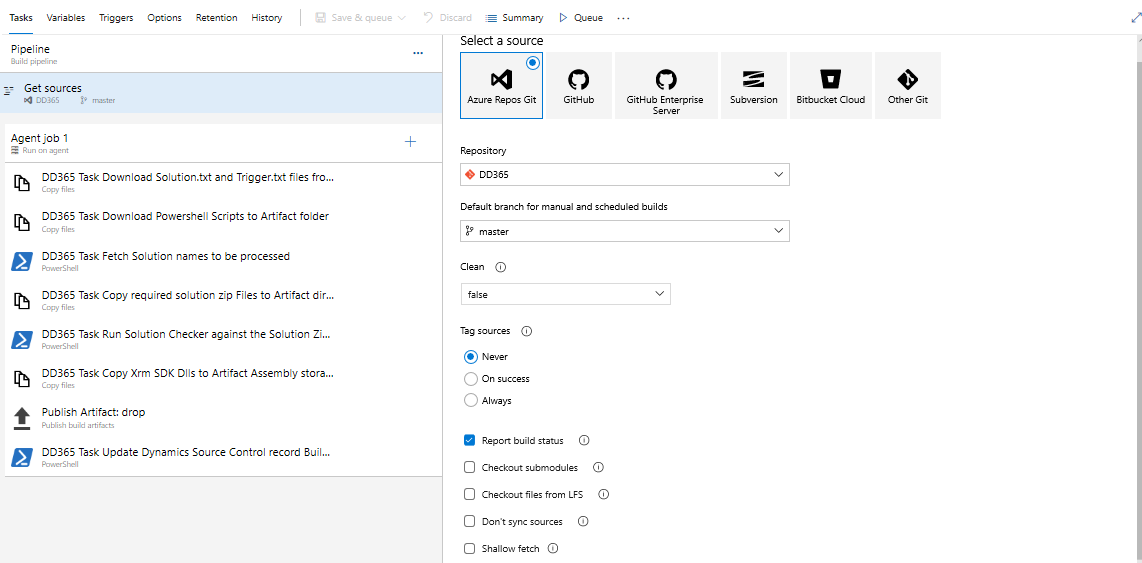




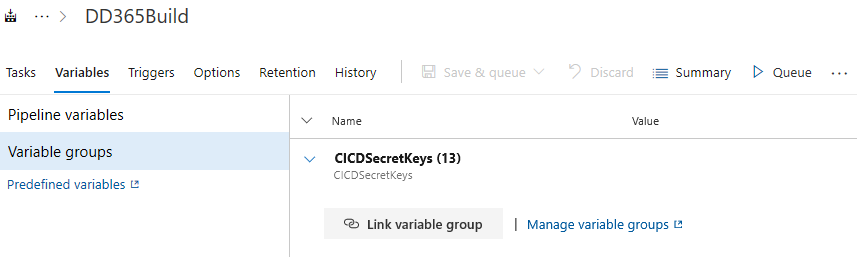
* + After importing build json, give build pipeline name and assign agent pool and agent specification.



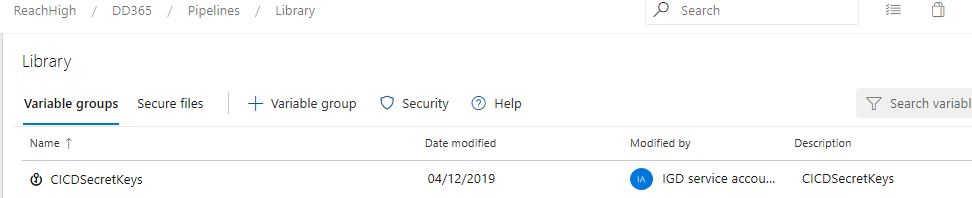
* + Click on **Get sources**, configure the branch



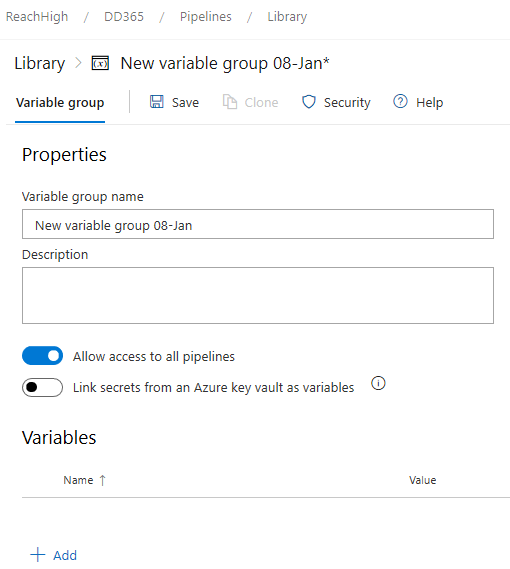
* + Click on variables, variables group and Remove existing variables group



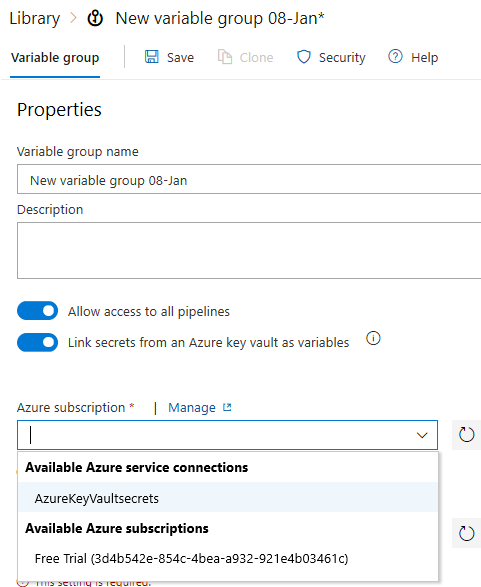
* + Click on manage variables group



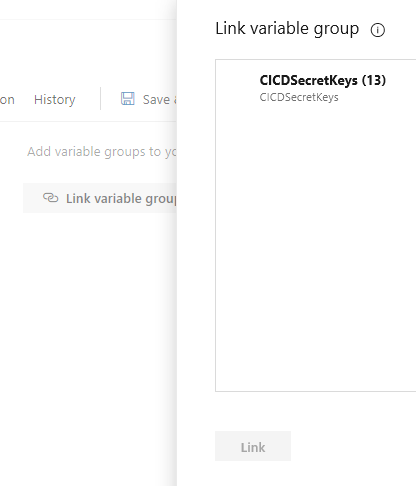
* + Click on new variable group



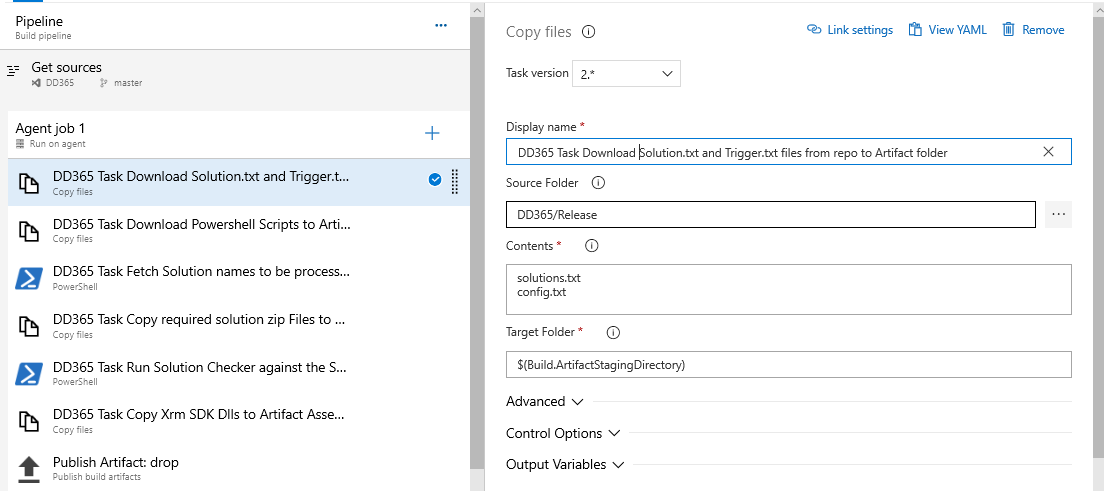
* + Give variable group name
  + Select link secrets and Select Azure Subscription and key vault name



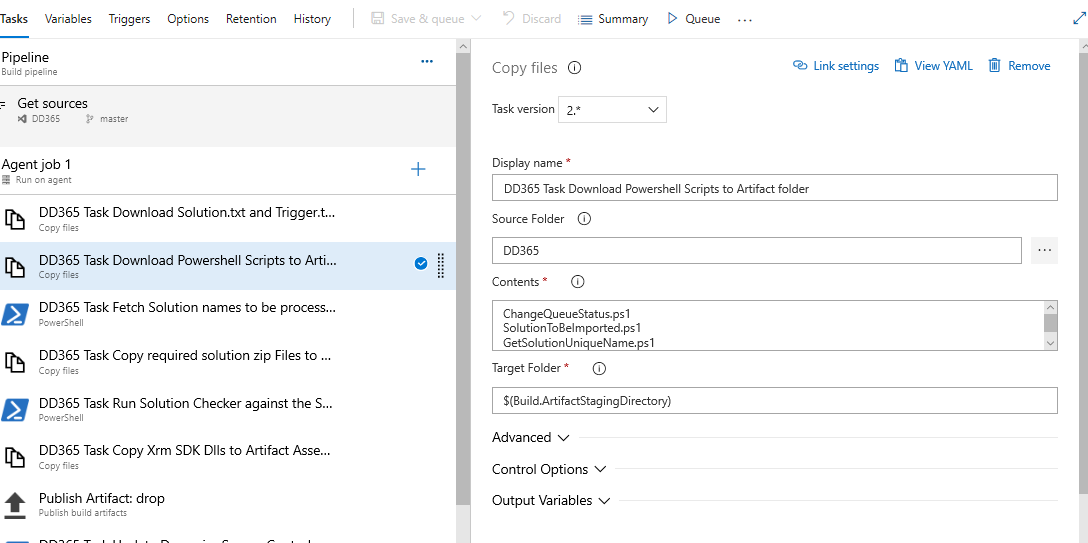
* + And click on add, add all key vault variables as mentioned in the DynamicsPOC\CRM Solution Manager\Release\WebJob\AzureKeyVaultNames.xlsx
  + After this come back to the build definition file click on variables group, link variables group, select the newly created variables group.



* + Under Agent job 1 - Need to check and update the task details in all the tasks .
  + Modify build definition paths from here on
    - Modify copy .txt from Main folder, click on source folder redirect to dd365/release
    - And do the similar thing for rest of the task replace DD365 with the path that is there in the repo
  + Modify copy zip files to from release source folder to the path where you want managed and unmanaged solution to be deployed
  + Below task to download the .txt files from the given path .



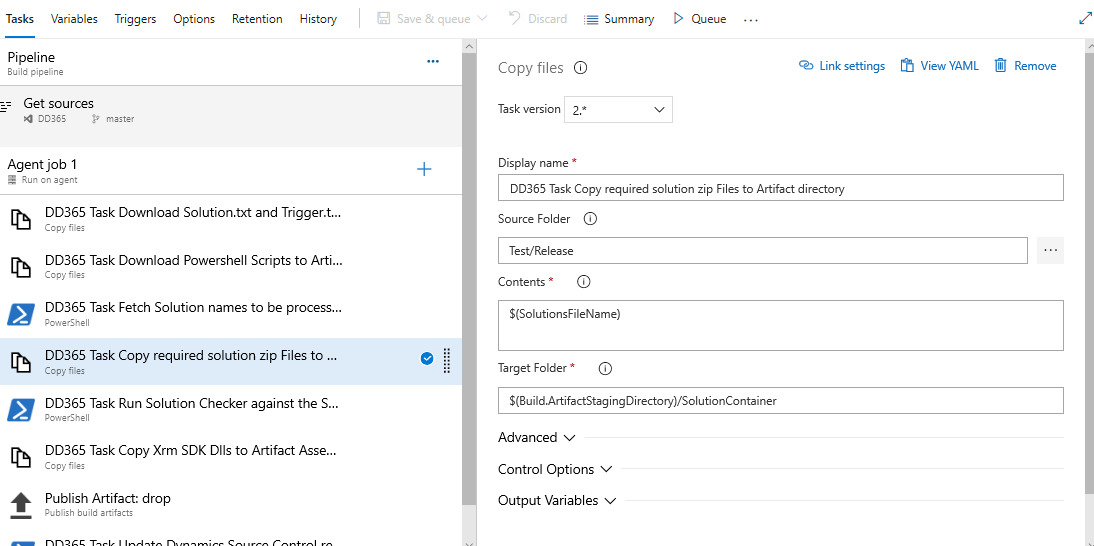
Below task to download .Ps1 files from repository



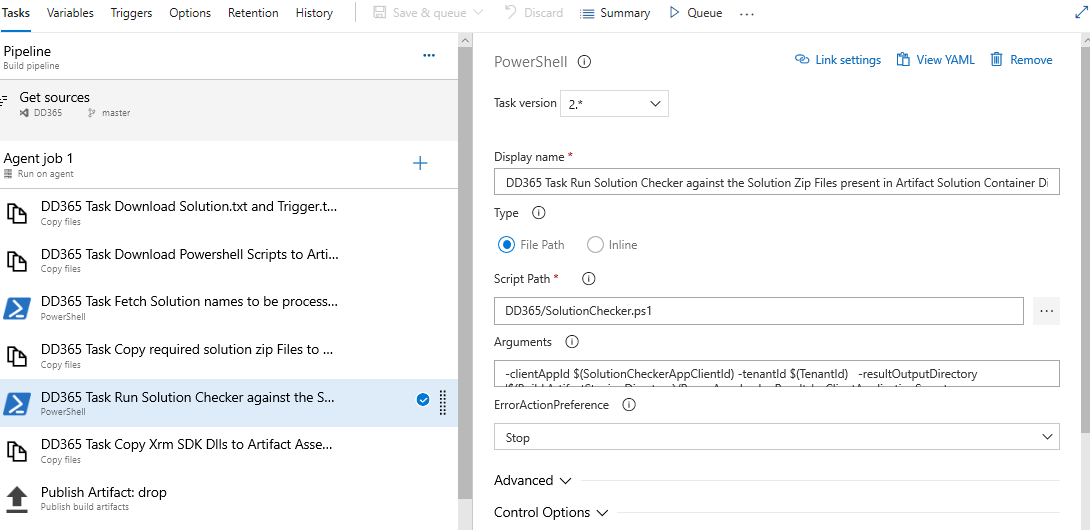
Below task to run the PowerShell script to get the solution names from .txt files from given path .



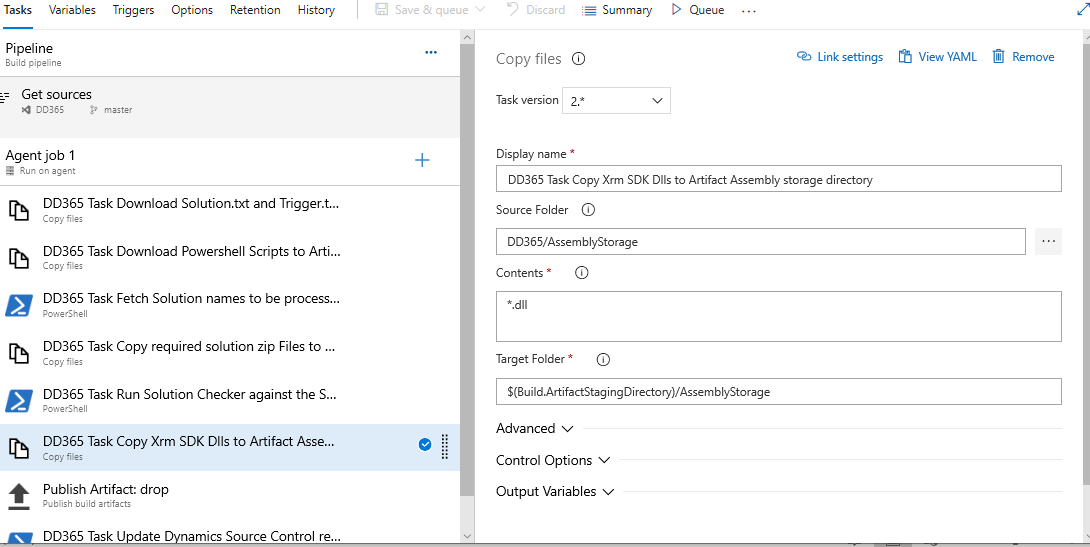
Below task to copy the solution files from repository to Artifact directory



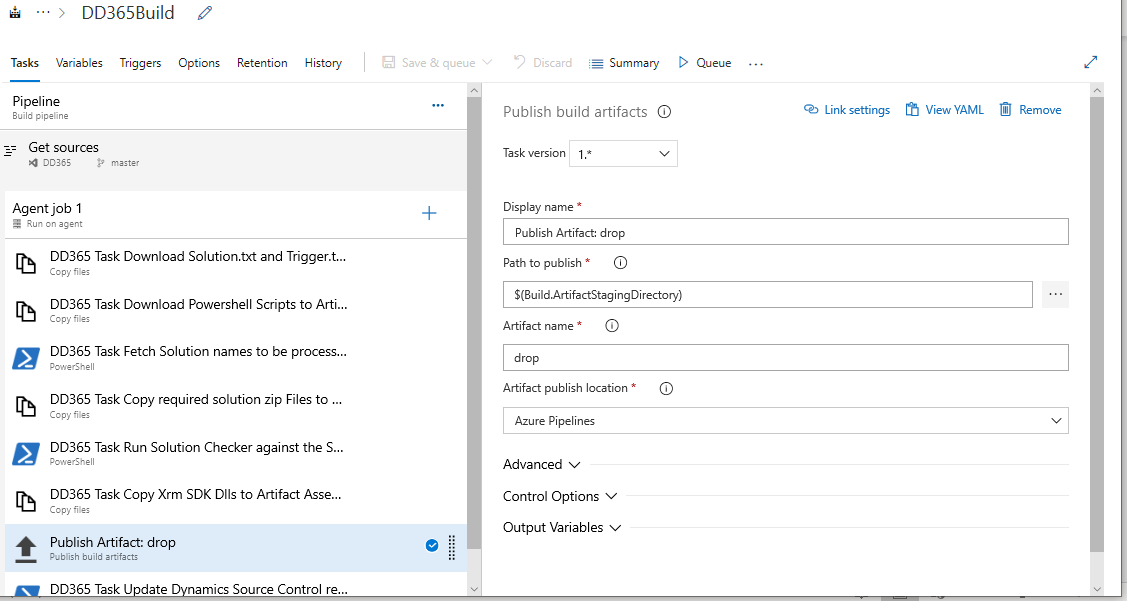
Below task to run Solution Checker against the Solution Zip Files present in Artifact Solution Container Directory



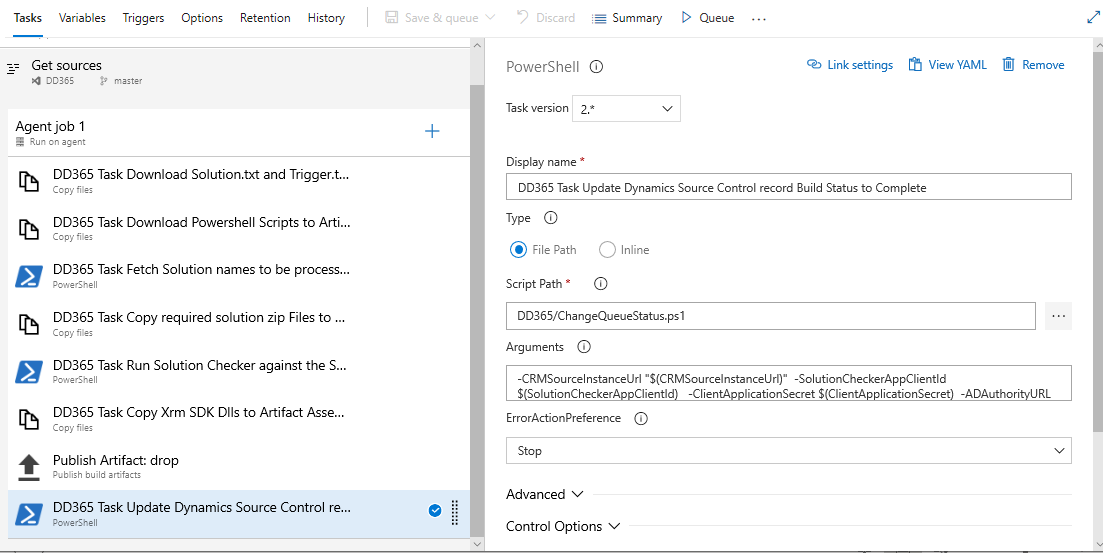
Below task to copy Xrm SDK Dlls to Artifact Assembly storage directory



Below task to drop all copied file to artefact directory location



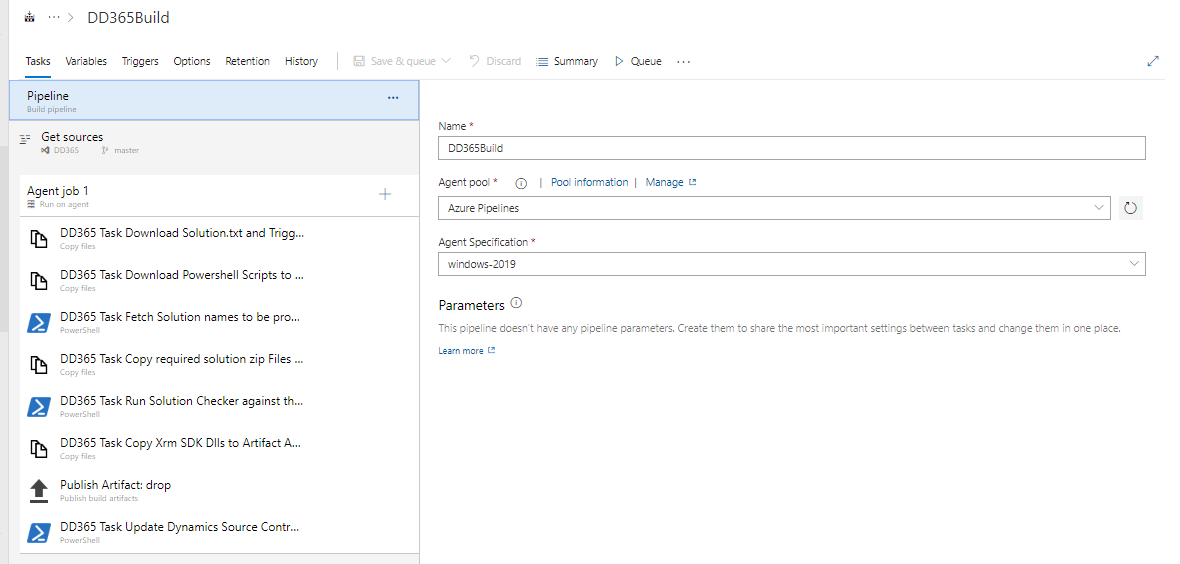
Below task to Update Dynamics Source Control Record Build Status to Complete



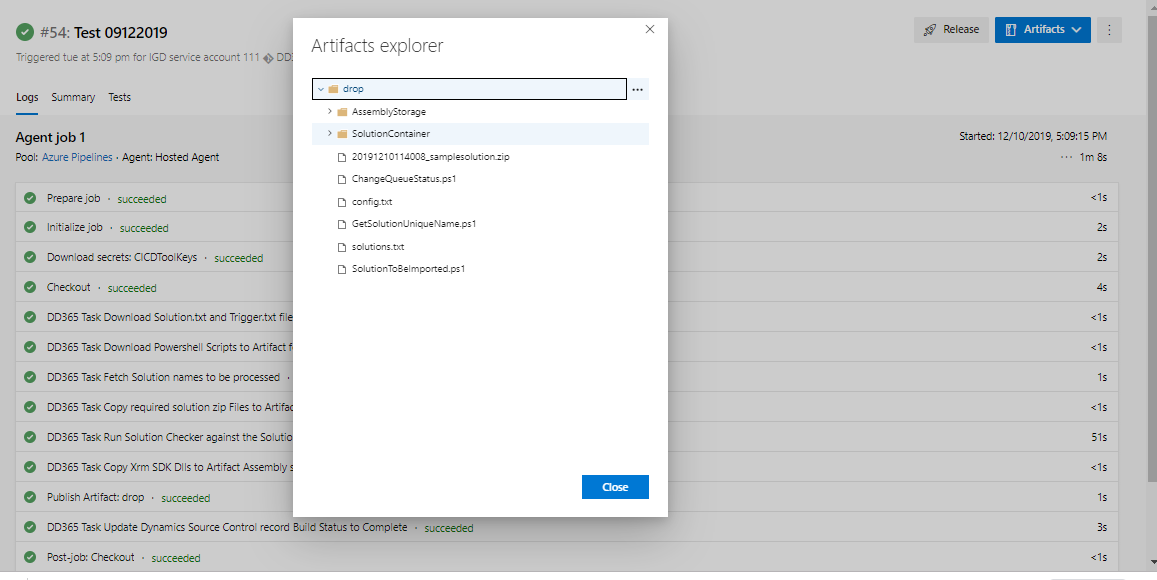
* + Update the trigger points   Goto triggers in build pipeline select Type and Branch selection
  + And Path Filters: select Type and Give Path specification as below



For more details refer the below image :



Build Artifact :



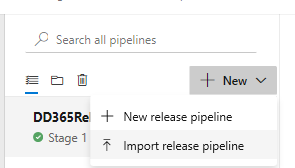
## **Configure Release Pipeline:**

Open folder DynamicsPOC\CRM Solution Manager\Release\Build and Release template

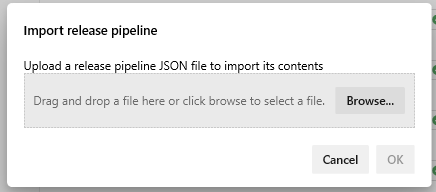


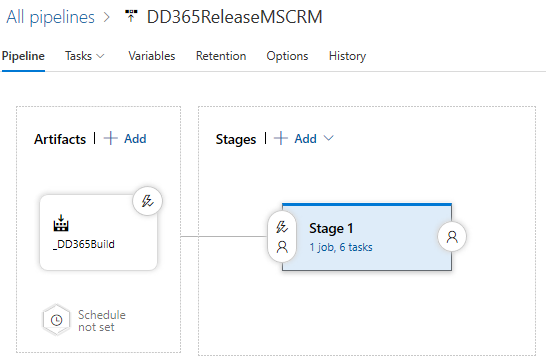
Go to Pipeline and Release pipeline in devops:

* + Import Release pipeline from release json.

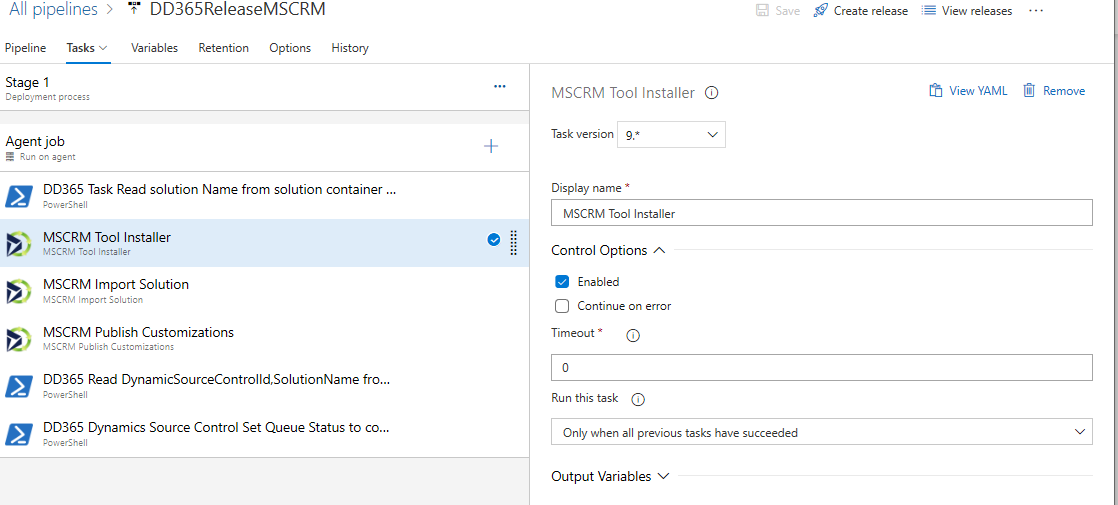


Click om Import release pipeline ,

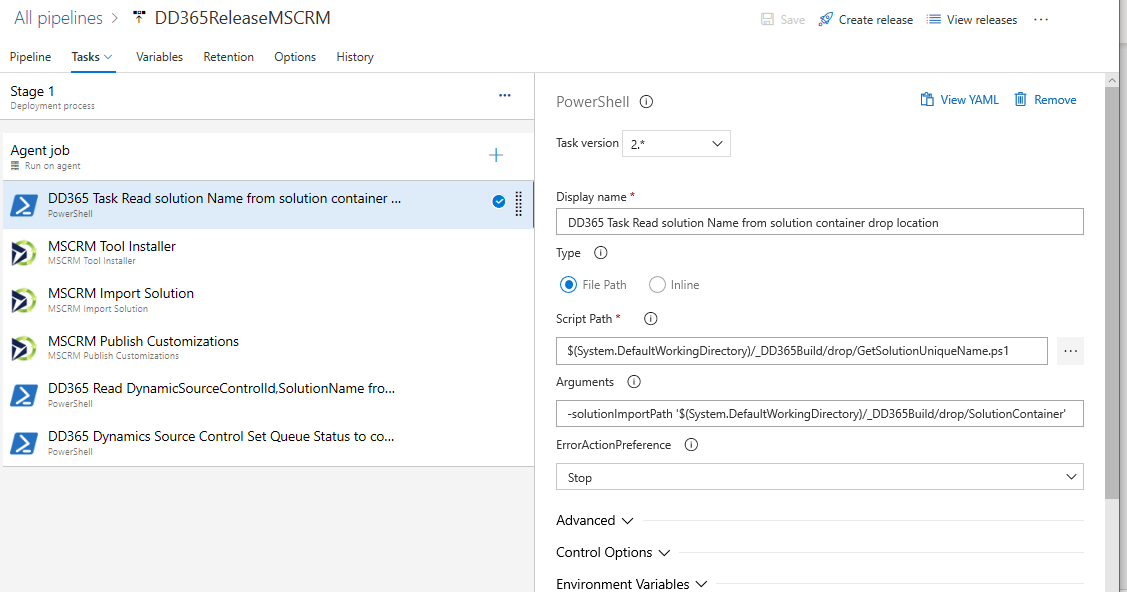




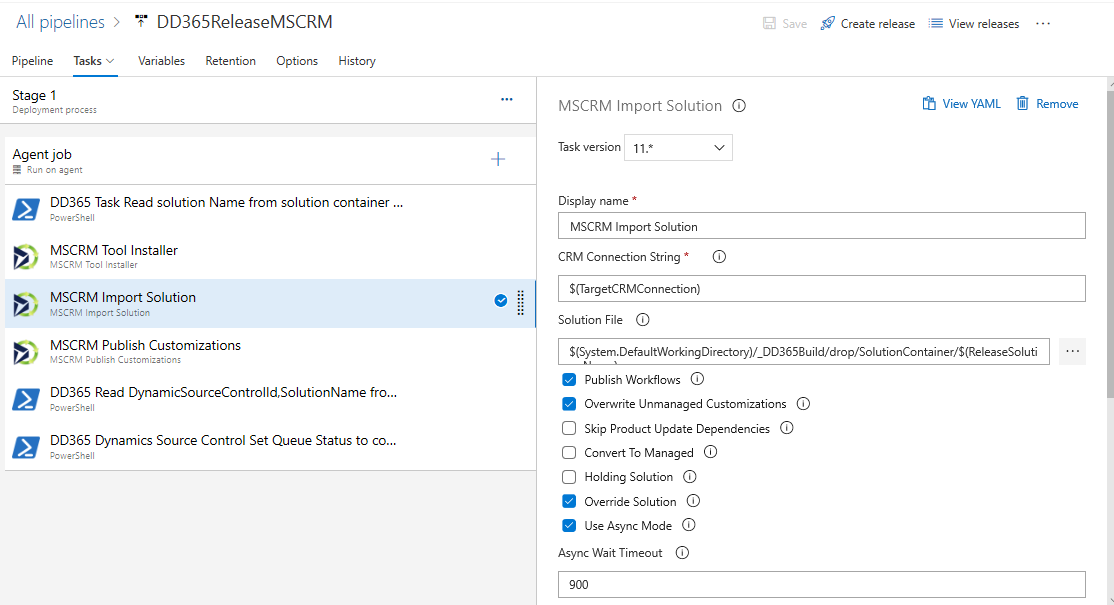
* + Check for MS CRM tool installer if not install it towards your organizations.
  + In MSCRM Import solution task need to change the solution file path to Build artefact and need to give Target CRM connection Key vault secret.



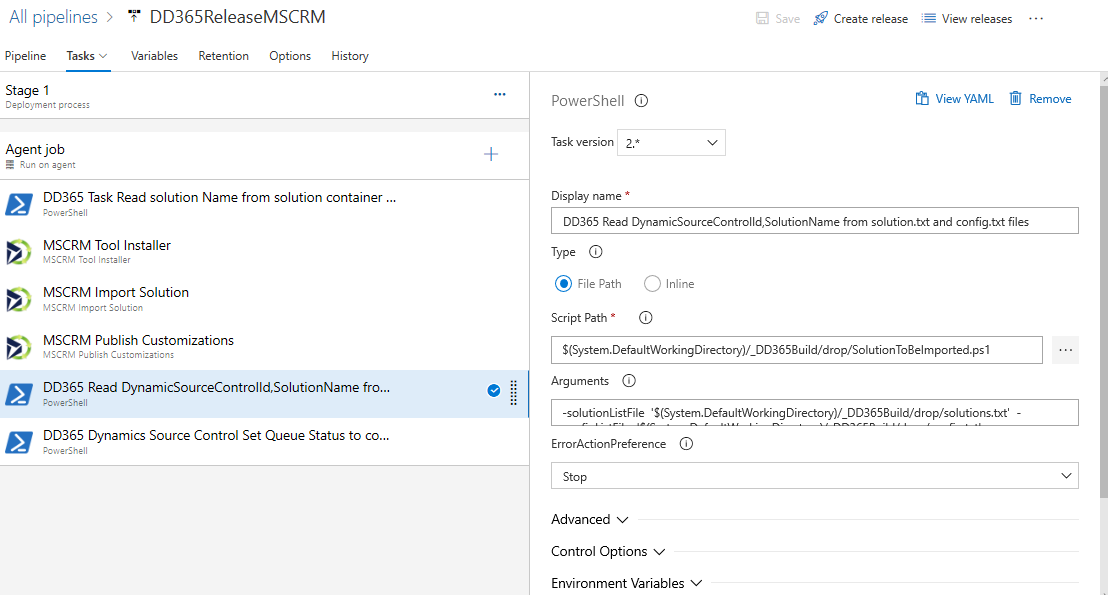
Below task to Read solution Name from solution container drop location



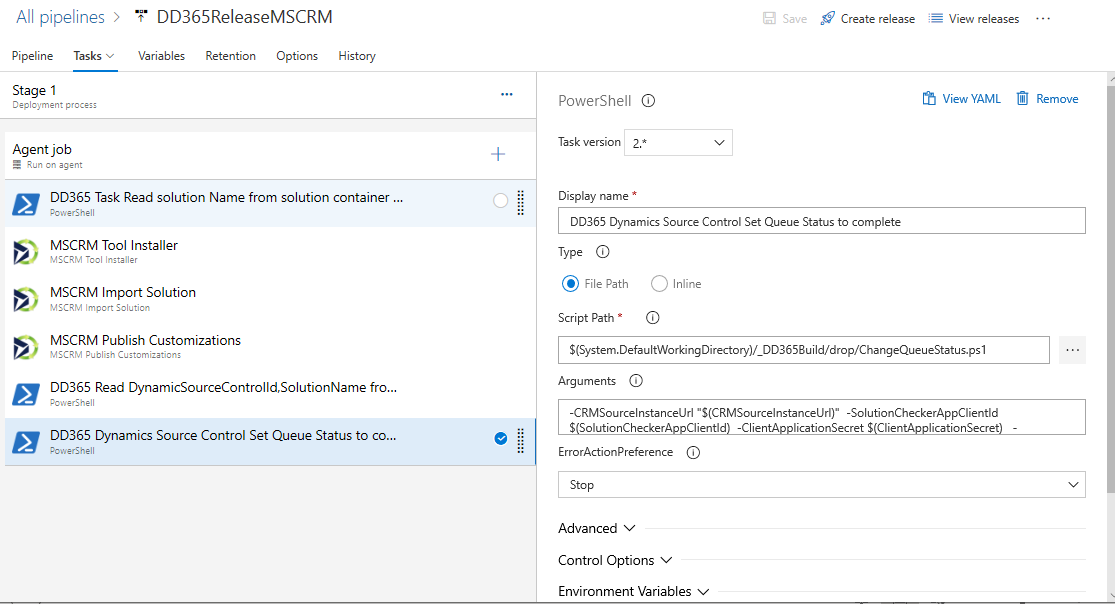
Below task to import solution to target environment



Below task to read DynamicSourceControlId, SolutionName from solution.txt and config.txt files

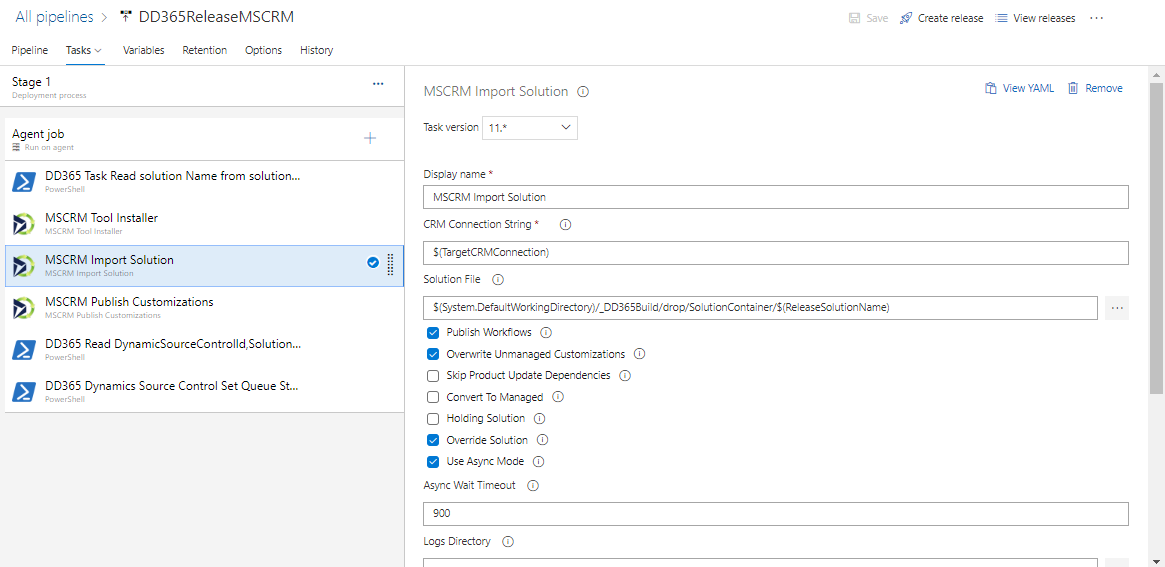


Below task to update Dynamics Source Control Set Queue Status to complete



* + In all above power shell tasks need to modify Script path to your folder path with PS file.
  + In all above power shell tasks argument section need to modify the build artifact path.

For more option please refer below image:



# **CRM instance configuration:**

Login into CRM Source instance:

* + Go to CRM Source instance Settings.
  + Click on Solutions and import below solution



* + Go to settings and Configuration settings form🡪 update the repository configurations, use below excel for the reference .



* + Add User with above Azure Active directory Application Id , save it .
  + Settings 🡪 Security🡪 Users 🡪 Select Application User 🡪 Create New Application User with Application ID
  + Assign Role to above user 🡪 go to manage role and Assign System Administrator, system Customizer roles to user.

# **Create MS power apps flow:**

Login to [https://india.flow.microsoft.com/](https://nam06.safelinks.protection.outlook.com/?url=https%3A%2F%2Findia.flow.microsoft.com%2F&data=02%7C01%7Cv-naryar%40microsoft.com%7C4b5fb46a3d30463e4ed208d77179921f%7C72f988bf86f141af91ab2d7cd011db47%7C1%7C0%7C637102639310401466&sdata=wDDgZxRAQ809s14%2F2rOL%2BzlVdxayB7i9r6ia%2BPmaI8A%3D&reserved=0)

* + Go to solutions open d365 solution
  + Go to Scheduled dd365 and Trigger Web Job When Record Is Created Or Updated flow and update below things for each step
    - Connection strings
    - Azure function URL
    - In the Http Action check and update the Azure function url with Query string parameters .

**URL** : <https://cicdazurefunction.azurewebsites.net/api/AzureFunction?D365=CRMSourceServiceUrl&D365INST=CRMSourceInstanceUrl&GU=GitUserName&GP=GitPassword&TU=TFSUser&TP=TFSPassword&AR=WEB>

**Headers** :

x-functions-key  : select the value of the Azure Function master key.

(**this is a above copied master key from Azure Function** )

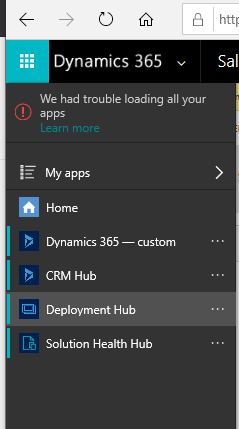
* Configure environment in the trigger action and Save it.

# **Tool Execution Steps and Features:**

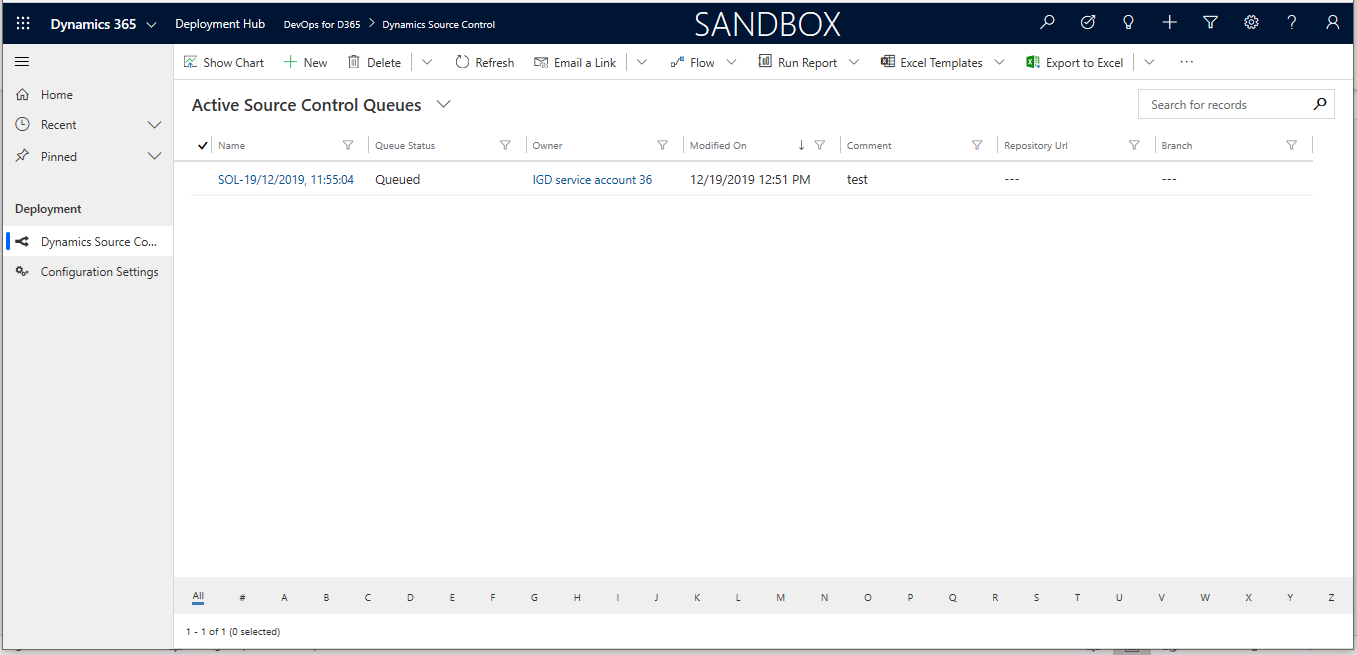
After doing all above set up tool need to be executed in the below order:

1.Create a record in Dynamic source control entity from its form as shown below:

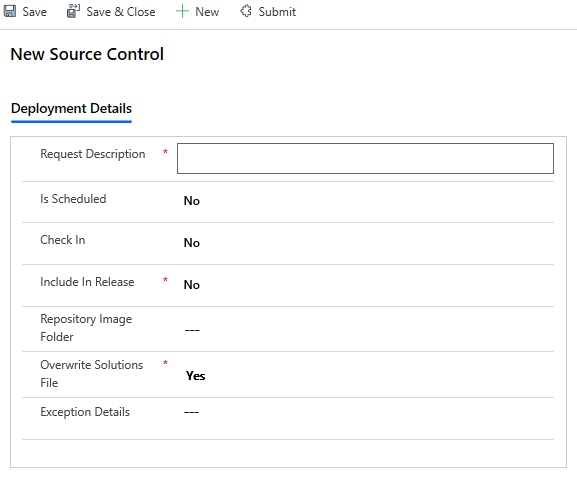
* Login to CRM source instance and navigate to Deployment Hub section:



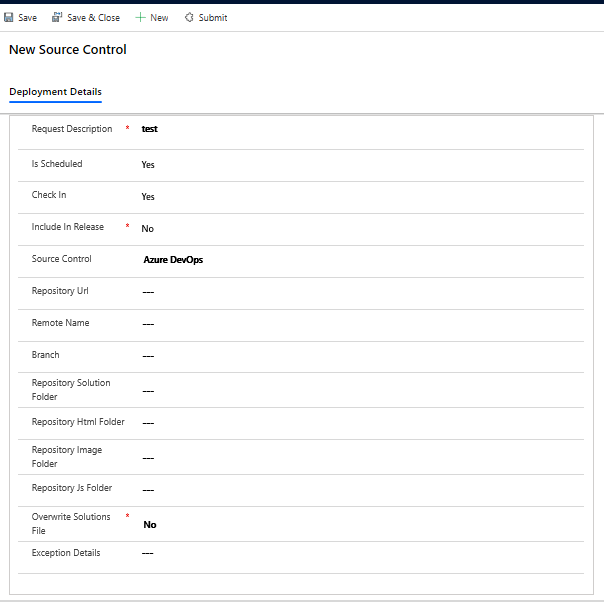
* Deployment Hub contains Dynamic source control and Configuration settings tabs:



* Create a New Dynamic source control record, click on **+New** link in the Dynamic source control form**.**

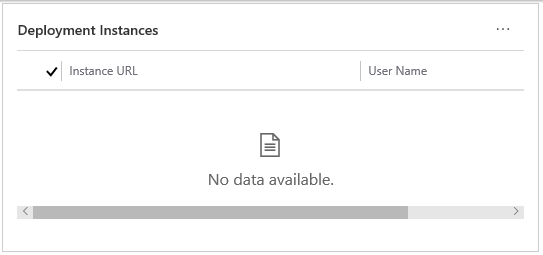


And

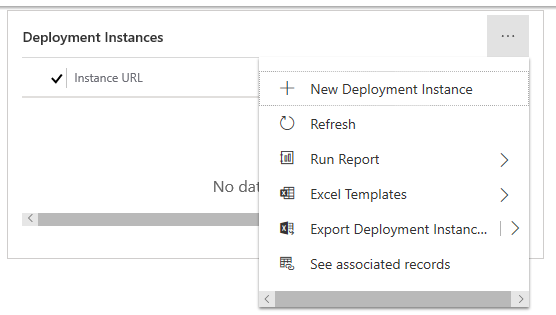


* **Request Description:** Need to give deployment description text.
* **Is Scheduled:** If Client project team want to do scheduled deployment then they need to select it as Yes, so that in Scheduled Microsoft flow they need to give the Time interval in Minutes or Hours or Days wise to run the deployment .
* **Check In:** if client project want to only check-in the solution to repository then they need to select it as Yes, so that it will commit all the solution details and data to repository .
* **Source Control:** If Check-in is **Yes** the this field will be shown in the UI and here we need to select the project source control to check-in the solution .
* **Repository Url:** If Check-in is **Yes** the this field will be shown in the UI and here we need to enter the project source control repository Url to check-in the solution .
* **Remote Name:** If Check-in is **Yes** the this field will be shown in the UI and here we need to enter origin and it will be used in api Url.
* **Branch:** If Check-in is **Yes** the this field will be shown in the UI and here we need to enter project branch name.
* **Repository Solution Folder:** If Check-in is **Yes** the this field will be shown in the UI and here we need to enter solution folder repository path to store the solution zip files.
* **Repository Html Folder: :** If Check-in is **Yes** the this field will be shown in the UI and here we need to enter Html folder repository path to store the solution Html web resources.
* **Repository Image Folder:** If Check-in is **Yes** the this field will be shown in the UI and here we need to enter Images folder repository path to store the solution images.
* **Repository Js Folder:** If Check-in is **Yes** the this field will be shown in the UI and here we need to enter JS folder repository path to store the solution Javascript files.
* **Include In Release:** if client project want to build and release the solution to Target instance then they need to select it as Yes, so that it will commit all the solution details and data to repository and trigger the build and release pipeline and solution will deploy to the target instance .
* **Overwrite Solutions File:** if client project team want to do multi solution import then they need to select it as **Yes.**
* **Exception Details:** Here tool will display the error details if any exception raised during the deployment process
* **Queue Status:** After saved the details then it will appear on the screen with **Draft** status .

After giving above details need to save and now Deployment instance details grid will be shown and here we can add the direct deployment details .



And click on **…** and



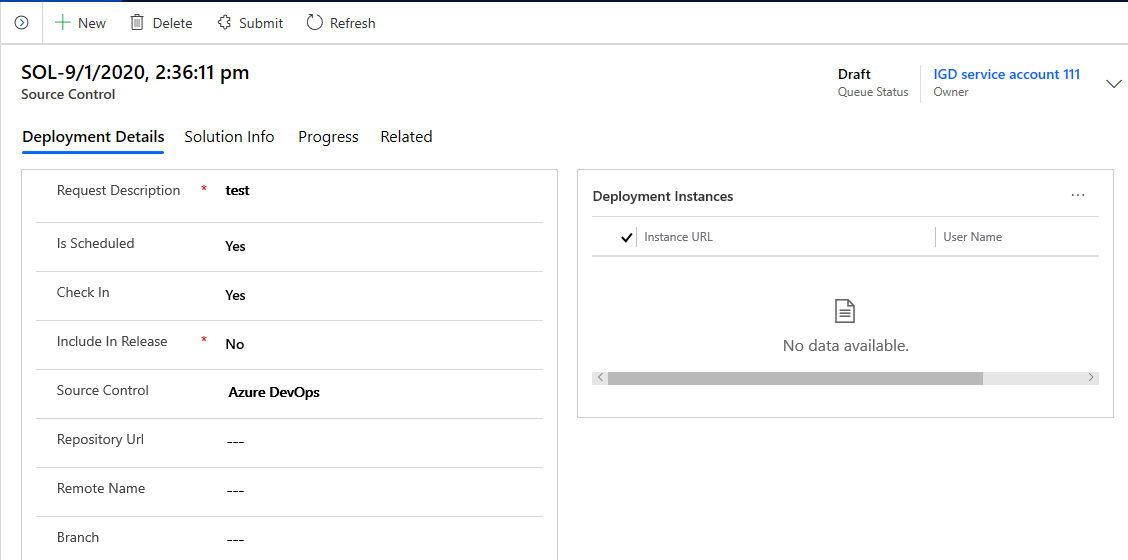
And click on +New Deployment instance in above screen



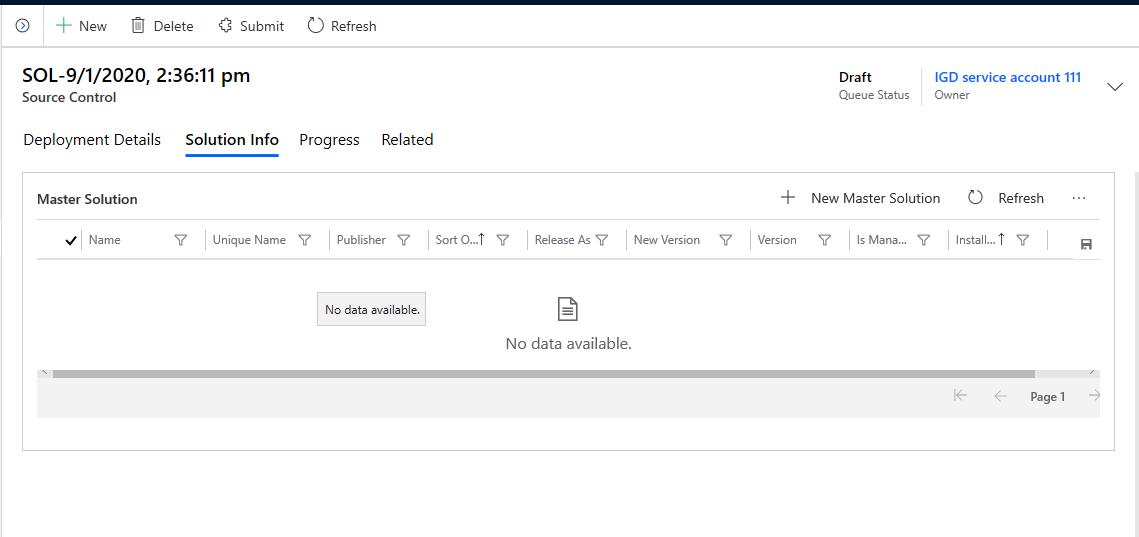
Here,

* **Owner:** Here signed in user details will be shown as default.
* **Instance URL:** we need to enter the Target CRM instance URL.
* **User Name:** we need to enter Target instance login user name.
* **Import :** we need to select yes if we are importing solution to target instance else No.
* **Check Dependency :** If Yes selected ,it will compare all components in target CRM instance it will log the all missing dependency components .
* **Clear Password:** It is YES then after deployment successful process we are clearing the password from back end entity.
* **Password:** we need to enterTarget CRM instance password.

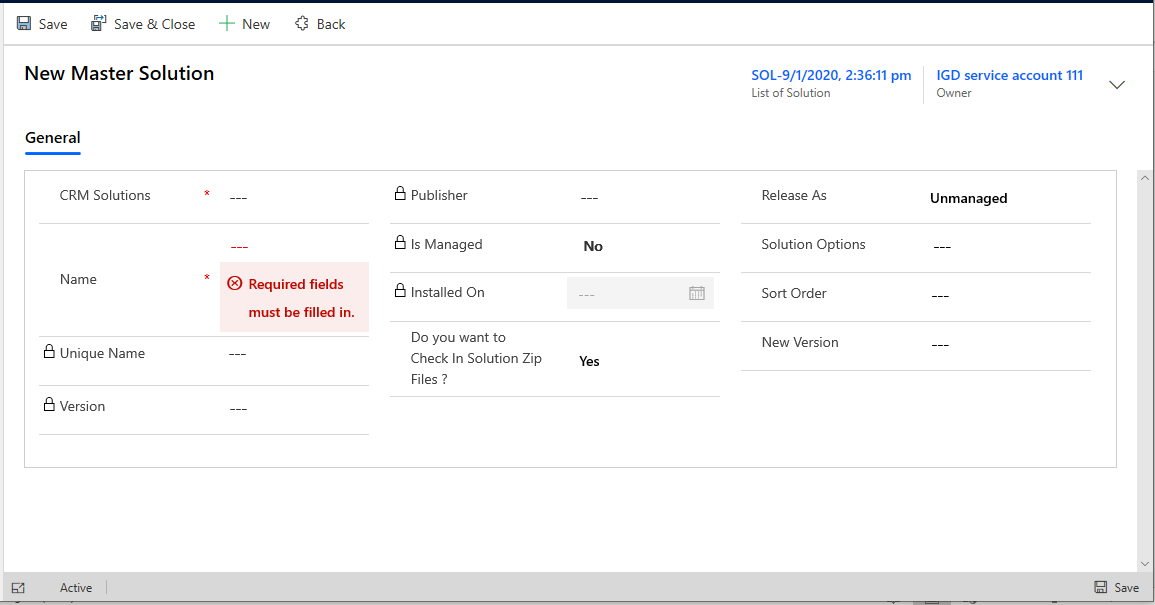
After saving the above details navigate to dynamic source control form then



Click on **Solution info** Tab

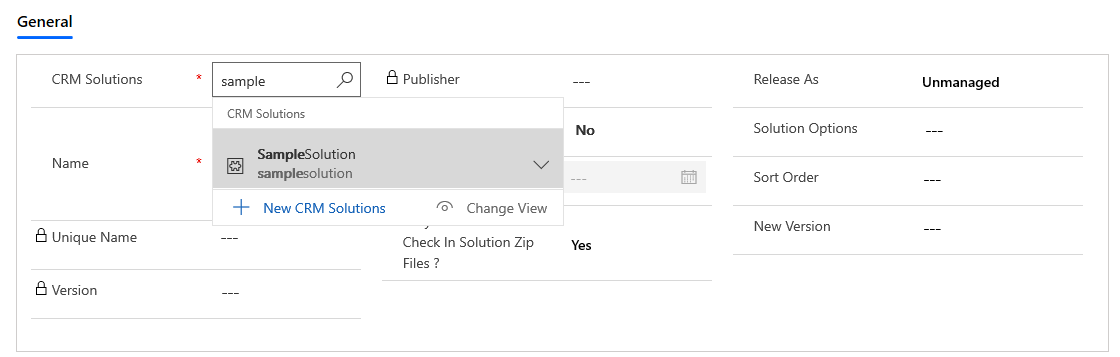


Click on +New Master solution button

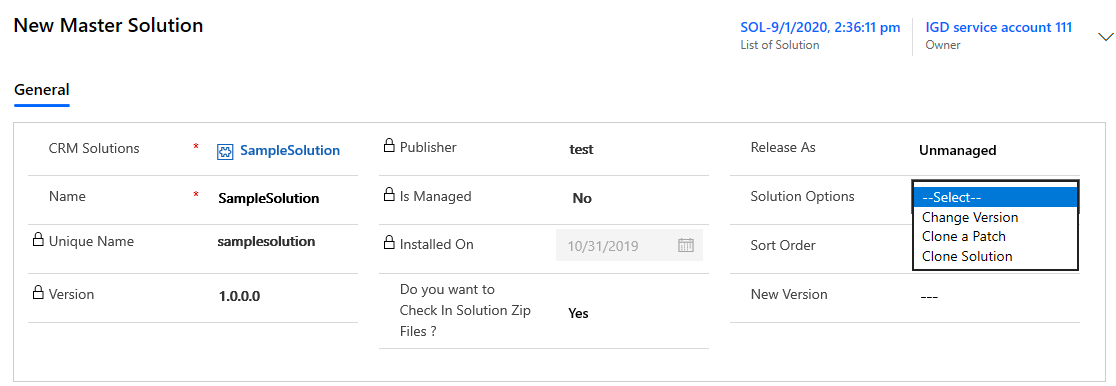


Here,

Select the Master Solution which you want to deploy it to Target instance .



Here we can change the solution options as shown in below screen .

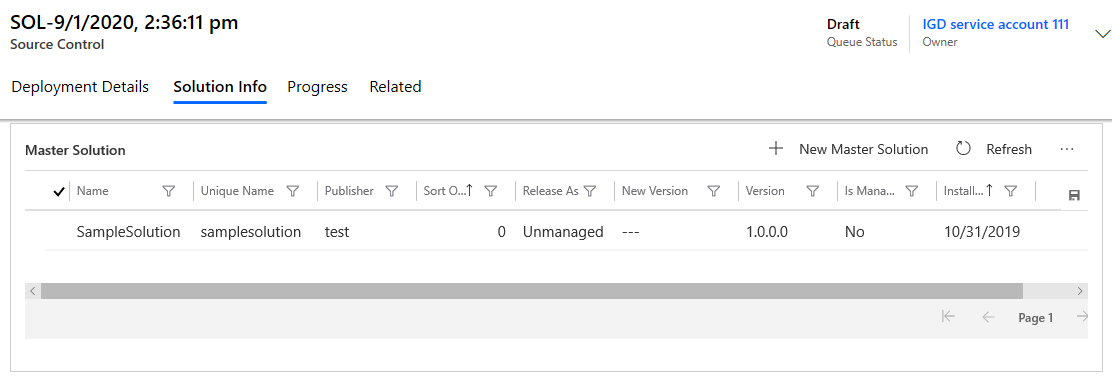


We will give sort number in sort Order to deploy in the specified order .

And we can specify the number in version number :

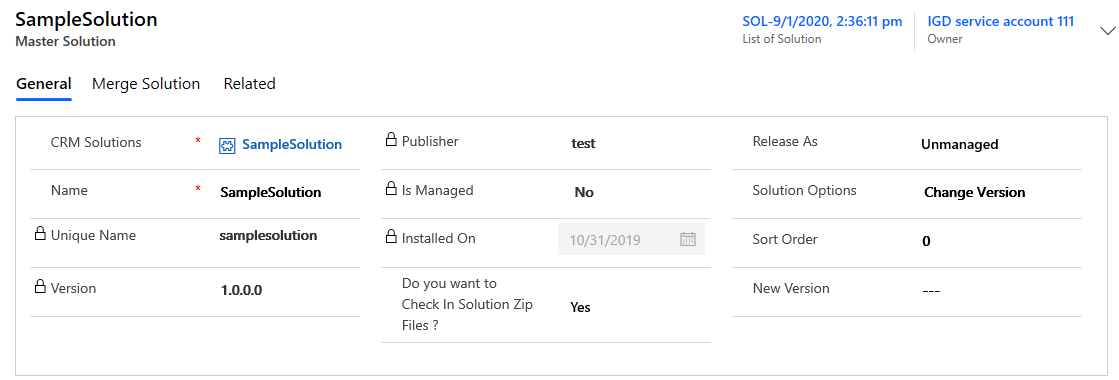


Click on Save and go back to Dynamic source control screen and in solution Info tab,

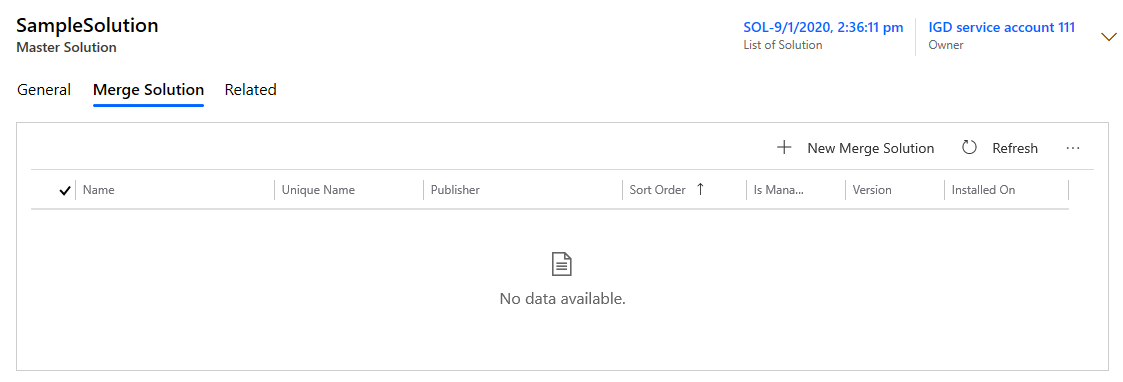


In the above manner we can add multiple master solutions to deploy to target instance .

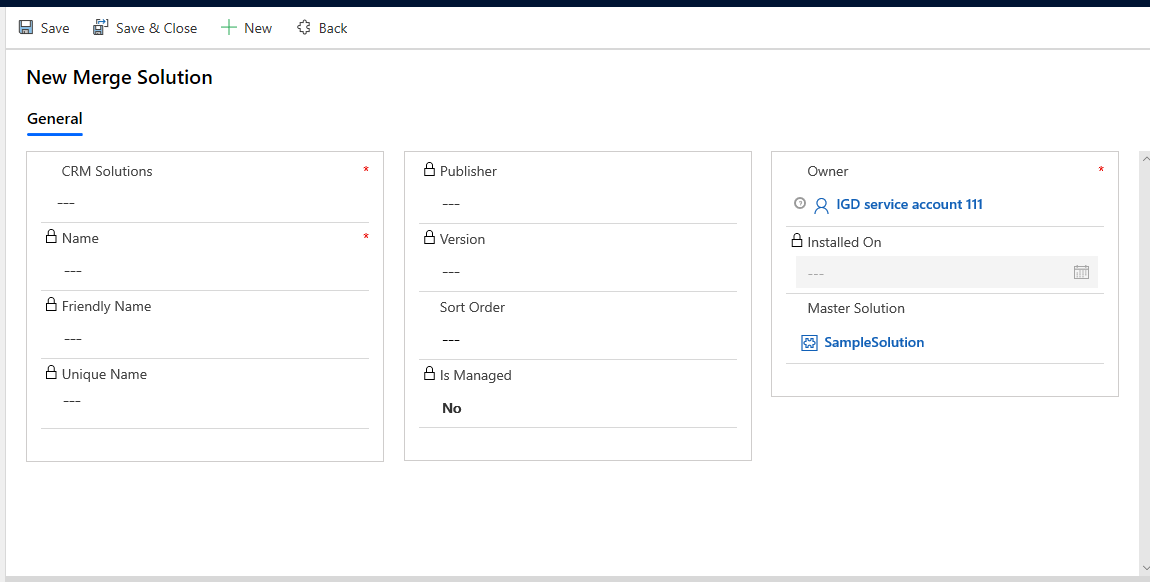
Click on the Master solution record and open it ,



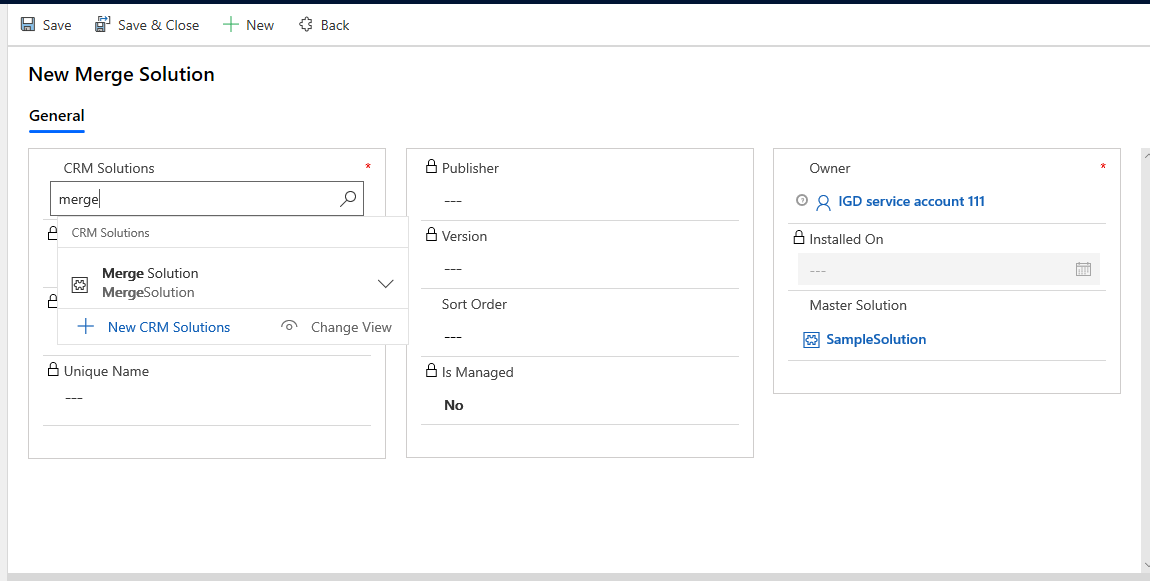
Click on Merge solution tab to Add the Child or merge solution that you want to merge with the master solution .



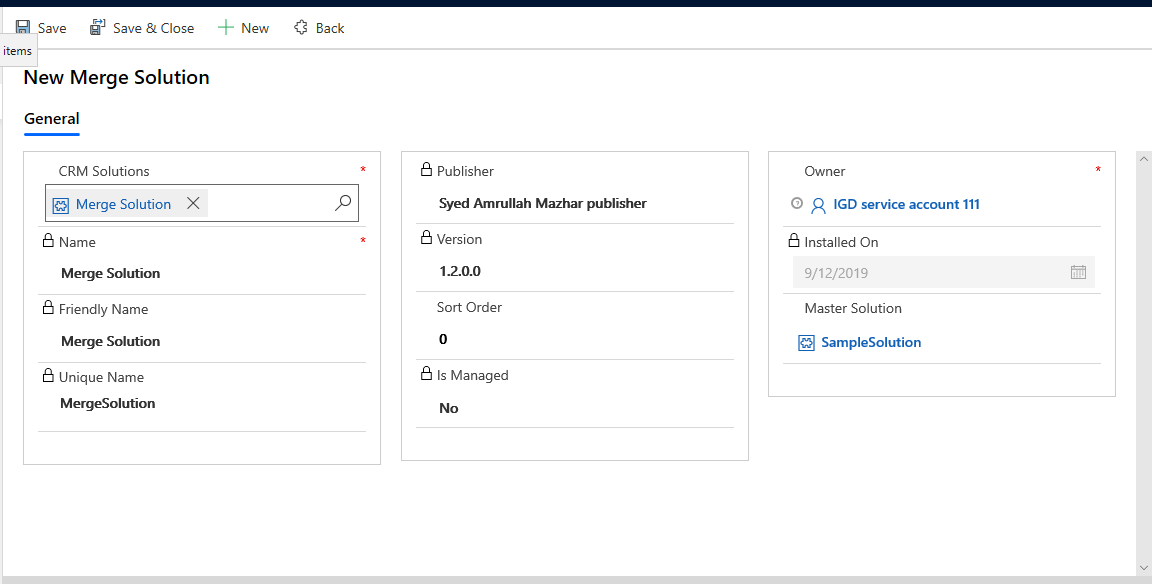
Click on + New Merge Solution button to Add merge solutions.



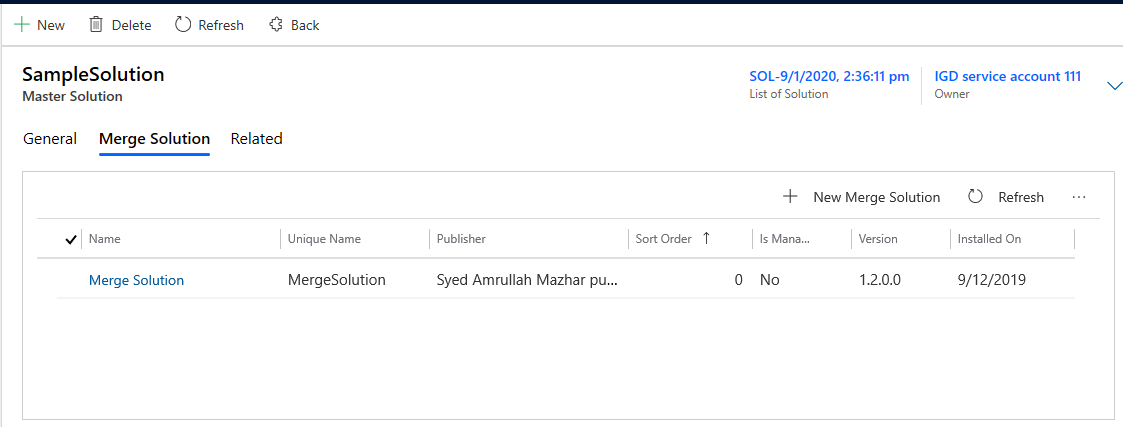
Select Merge solution in the CRM solution look up field to merge it with master solution .



And here all merge solution component will be merges with the master solution and as a single solution it will deploy to the target instance . Here we can mention the Sort number to get executed in the given order .

****

Click on save and close then,



Here we can add number of merge solution to merge with master solution .

After Saving all above data we need to submit the record then the Que status will be change to the ‘**Queued**’.

Once the record status changed to **Queued** thenMicrosoft flow (in Turned On status )will trigger the azure function Which is hosted in azure .

Azure function will do below tasks:

* Connecting to Source instance and fetching the dynamic source control records which are in ‘Queued’ status .
* It will perform the merge operation from merge solution components to Master solution.
* It will save that result Master solution back to source instance.
* It will check for the direct deployment details is there then it will directly deploy it to target instance.
* It will get the new updated solution from source instance and extract the component and web resource from it and keep in the temp location.
* It will check for the condition like check in is Yes or not in the dynamic source control , if Yes it will Connect to repository and perform the commit Operation
* After committed all files to repository then there are .Txt files, here we are writing the solution names and source control Id.
* As we mentioned in the pipeline setup, the above txt files will help to trigger the build pile line once they got any changes, And here After completing build, we are updating the Queue status to build completed .
* After Build runs successfully, it will trigger the release pipeline, After completing the release we are updating the Queue status to release completed.
* In the above process we are logging all the meta data and logs to Process Tab web job logs section.
* If any Exception raised any of above stages, we are logging the exception details to logs.

