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

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

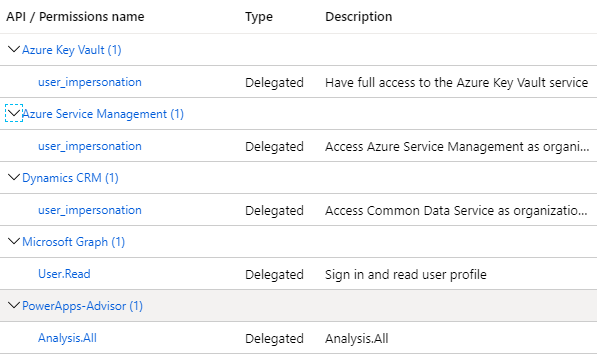
## **Azure Active Directory:**

Go to Azure active directory and register new application:

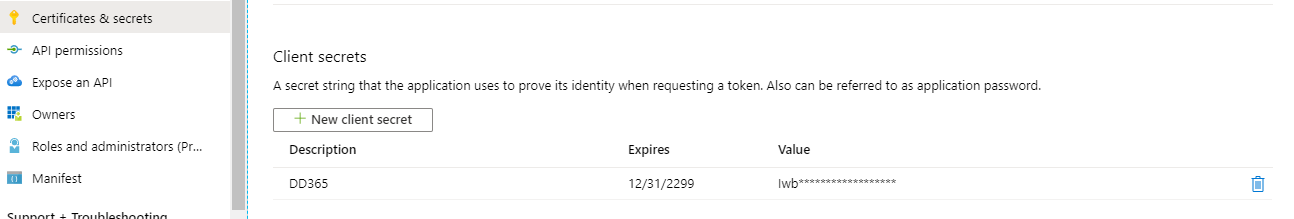
After Creating new application, Copy **Application Id** and **Tenant Id** to clipboard.

Give API permission to application, Click on API Permission:

* + - * Click on +Add permission
      * In Microsoft ApIs, 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.

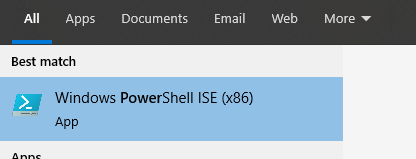


Create Azure storage Account, App service plan to deploy azure function in the portal.

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.

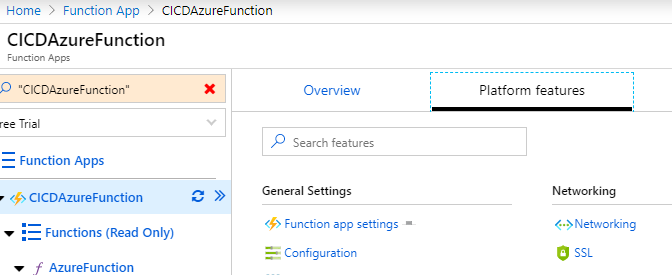
Now open PowerShell tool and run below CLI command with details :



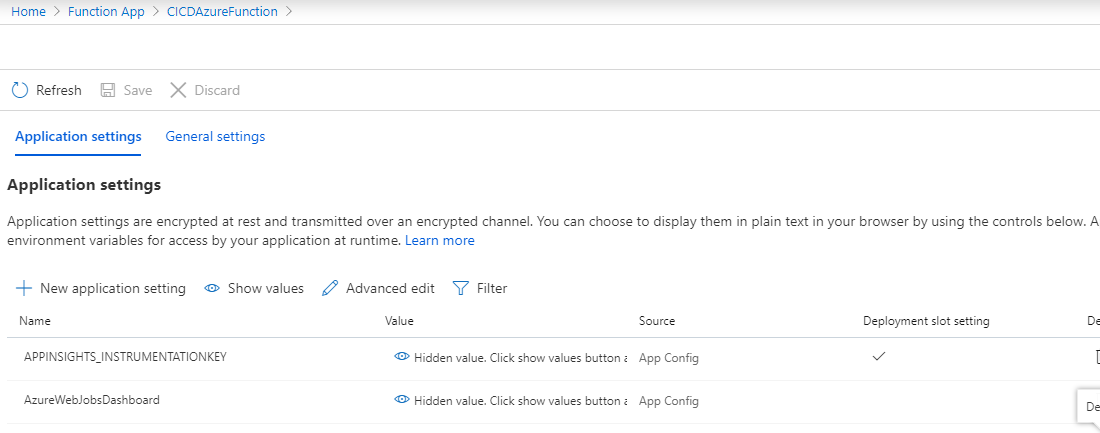
**az login  
az account set --subscription "subscriptonsname"  
az functionapp deployment source config-zip -g "resourcegrupo" -n "dd365compassazurefunciton" --src "E:\AllRepos\DynamicsPOC\CRM Solution Manager\Release\AzureFunction\CICDAzureFunction.zip"**

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

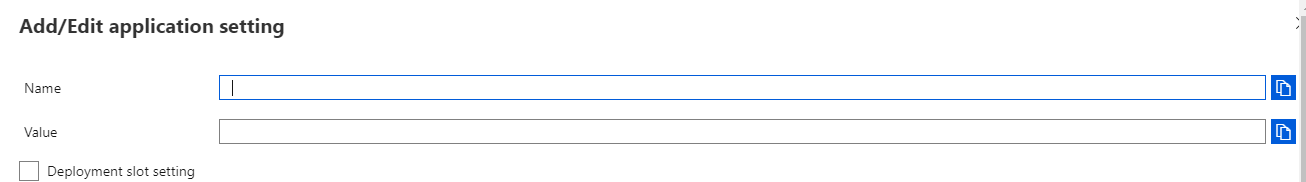
In the Function App , Go to Platform features :



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",

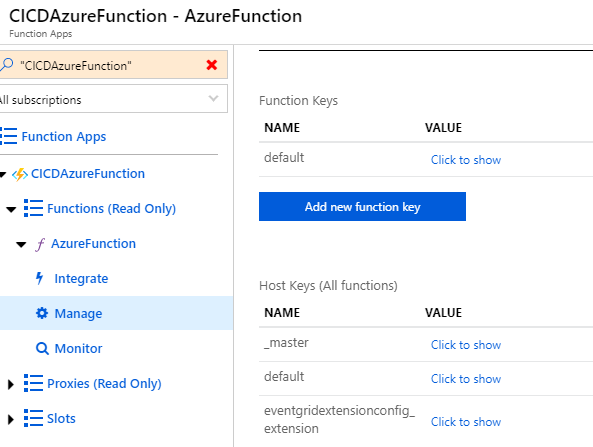
"TenantId": "your tenant Id",

"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:



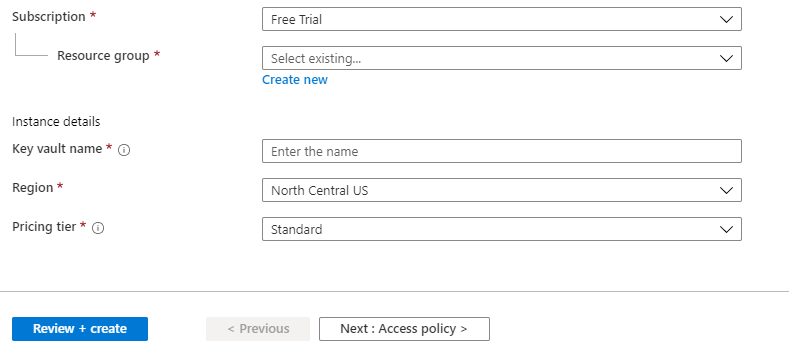
Click on \_Master **Click to show** link in the above screen:



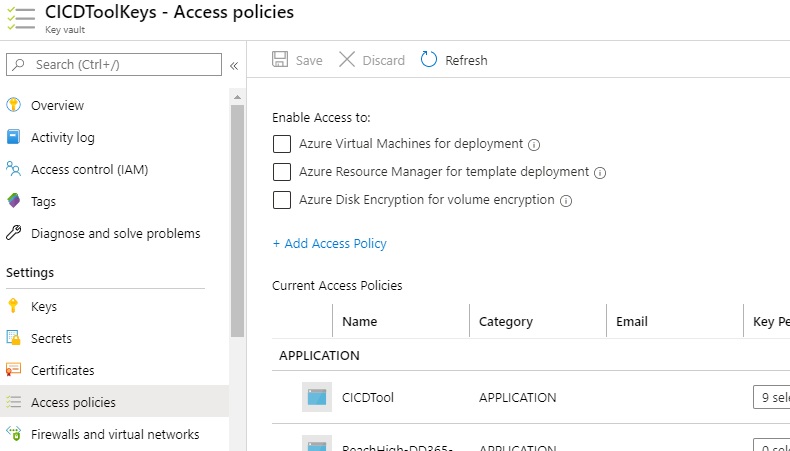
Click on copy link and copy master key to click board.

## **Azure Key vault creation:**

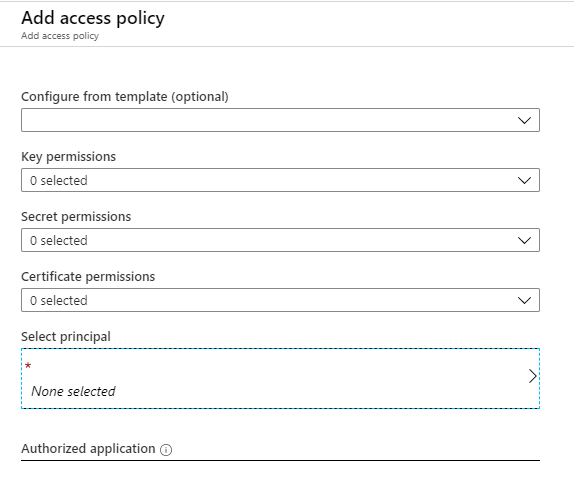
Go to Key vault and click on Add, give subscription ,Resource group, key vault name, Region, pricing tier and click on **Review+Create button .**



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



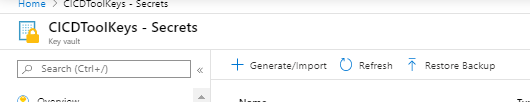
Now click on + Add Access Policy:



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



Now go to Secrets in the key vaults:



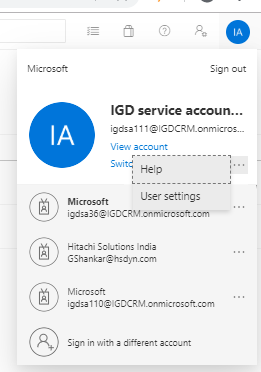
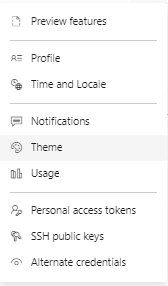
Click on +Generate/Import button and add the key secrets , please fallow the below excel to add the secrets .



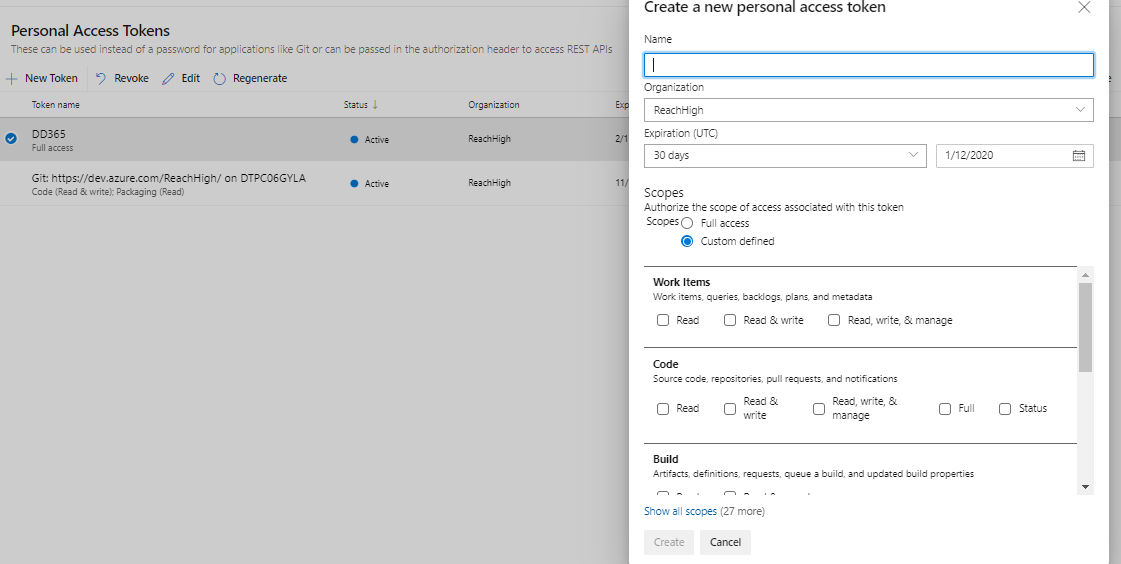
# **Devops Configuration:**

## **Personnel Access tokens** :

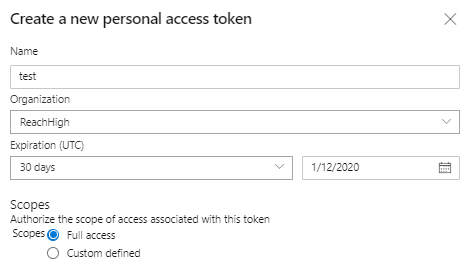
Login to devops (<https://dev.azure.com/>) and Create a PAT token, in the DevOps go to extreme right, select your user , click on …, click on User Settings, click Personnel Access Tokens, Generate New Token.

Create anew Access token:

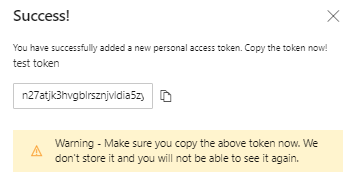


With Full Access:





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.

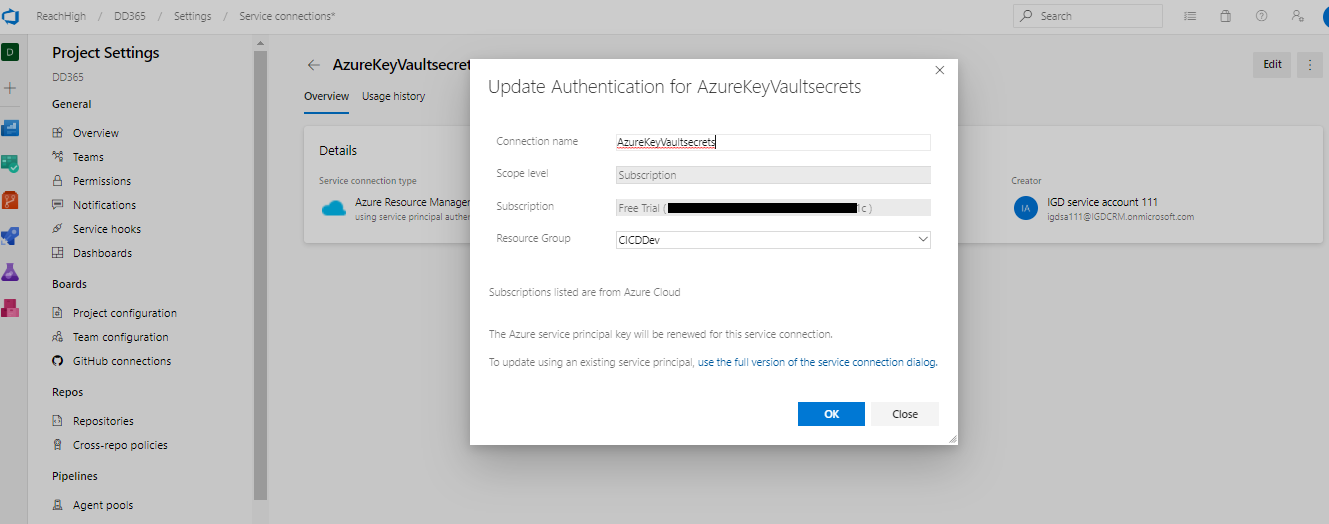
Configure Azure key Vault with the details as mentioned in the excel



## **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

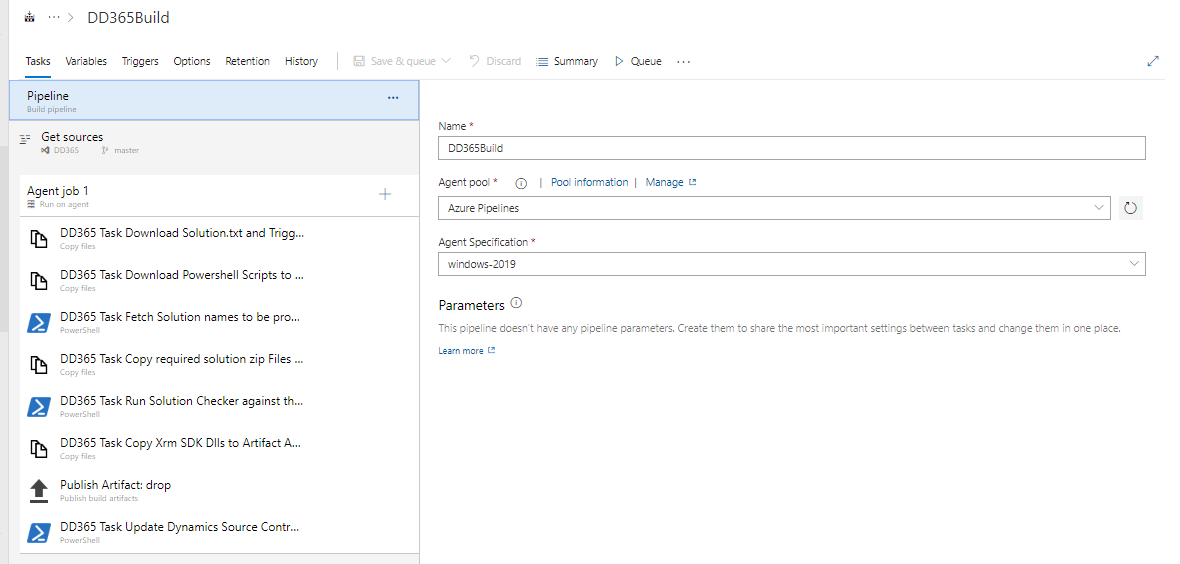


Go to Pipeline and build pipeline in devops :

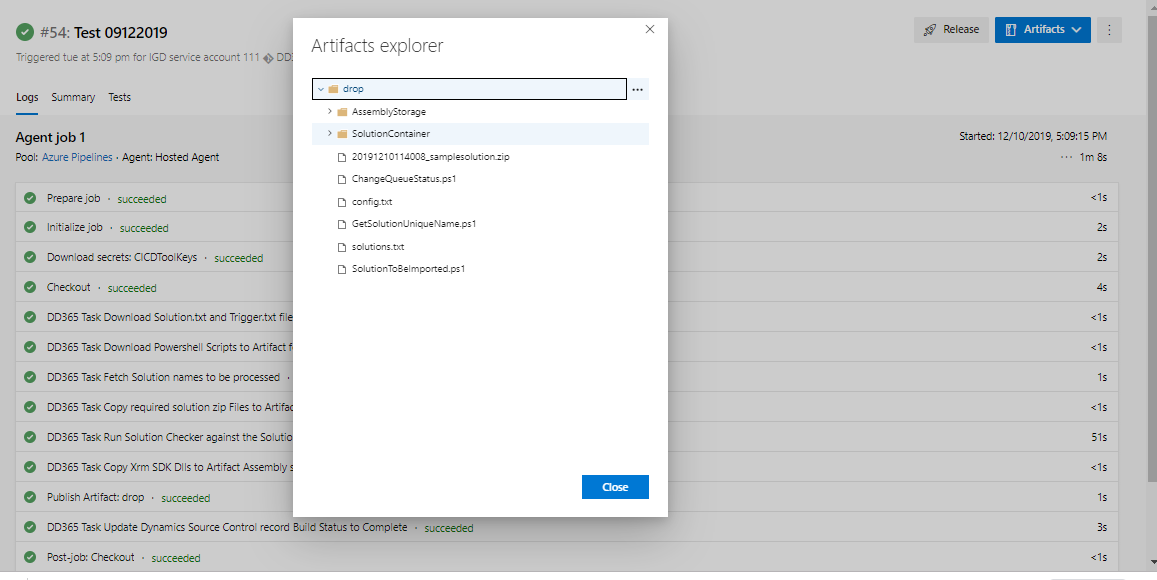
* + Click on build, click on new, click on import a pipeline and select the build json from above path
  + After importing build json , Give build pipeline name and assign agent pool and agent specification .
  + Click on get source , configure the branch
  + Click on variables, variables group and Remove 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.
  + 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
  + 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:**

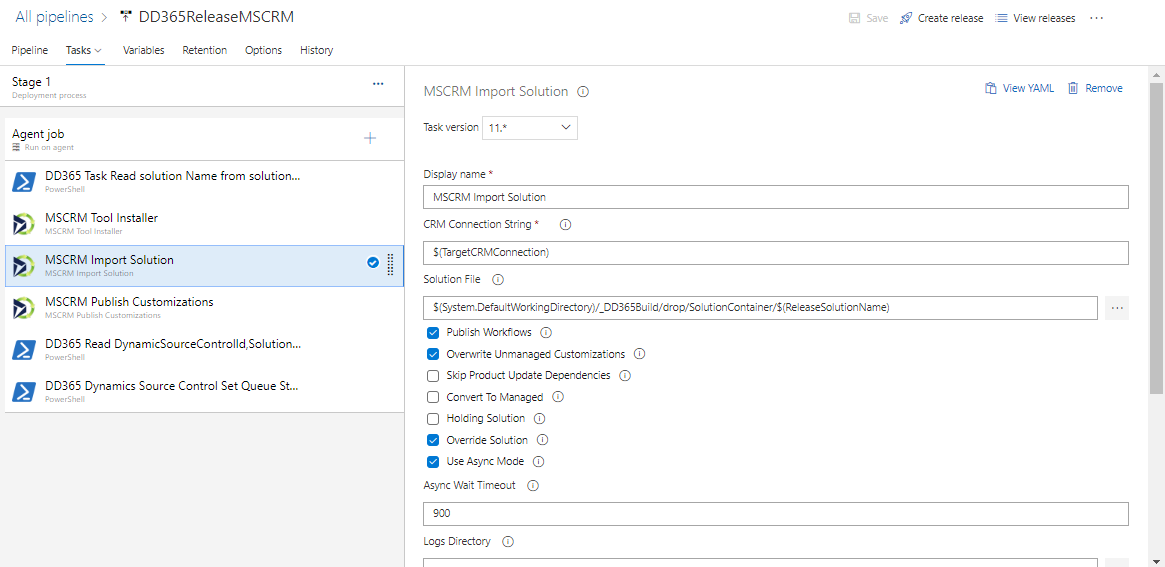
Open folder DynamicsPOC\CRM Solution Manager\Release\Build and Release template



Go to Pipeline and Release pipeline in devops:

* + Import Release pipeline from release json.
  + 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.
  + In power shell tasks need to modify Script path to your folder path with PS file.
  + In argument section of power shell tasks 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.