**CICD Build Process:**

Terra form extesnion :

https://github.com/hashicorp/vscode-terraform/releases/tag/v1.4.0

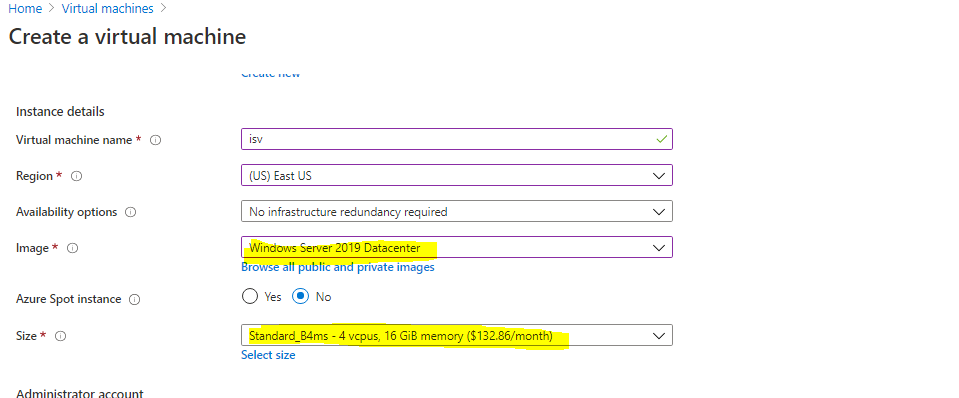
Terraform init

terraform plan

terrafrom apply

terrafrom destroy

* **Create Virtual Machines with size** **Standard\_B4ms(4cpus and 16GB RAM)**



* Login into above VM using crtl+r and type mstsc
* Open the powershell as adminstrator and need to install the choco by executing the below script

**Install Choco via powershell:**

Set-ExecutionPolicy Bypass -Scope Process -Force; [System.Net.ServicePointManager]::SecurityProtocol = [System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-Object System.Net.WebClient).DownloadString('https://chocolatey.org/install.ps1'))

choco install azure-cli

* Need to install the visualstudio2019buildtools in vm, run the below command in powershell

choco install visualstudio2019buildtools

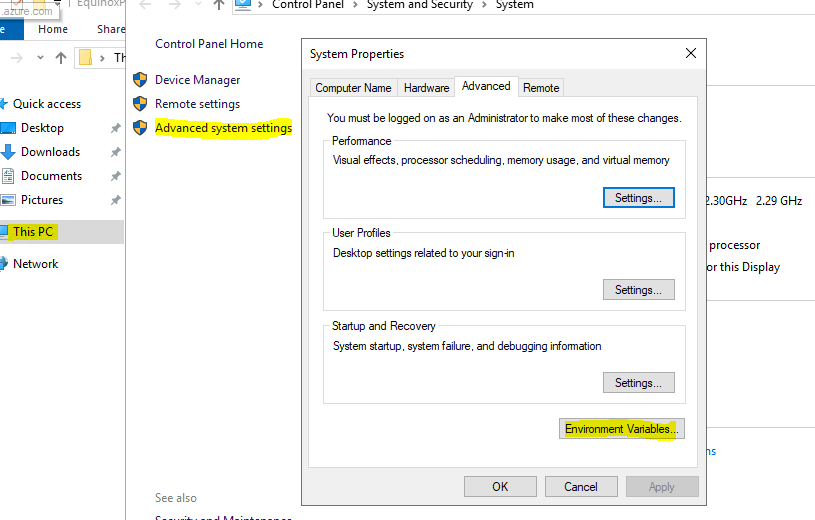
* After install, it will ask for reboot the vm, run the below command for reboot.

restart-computer

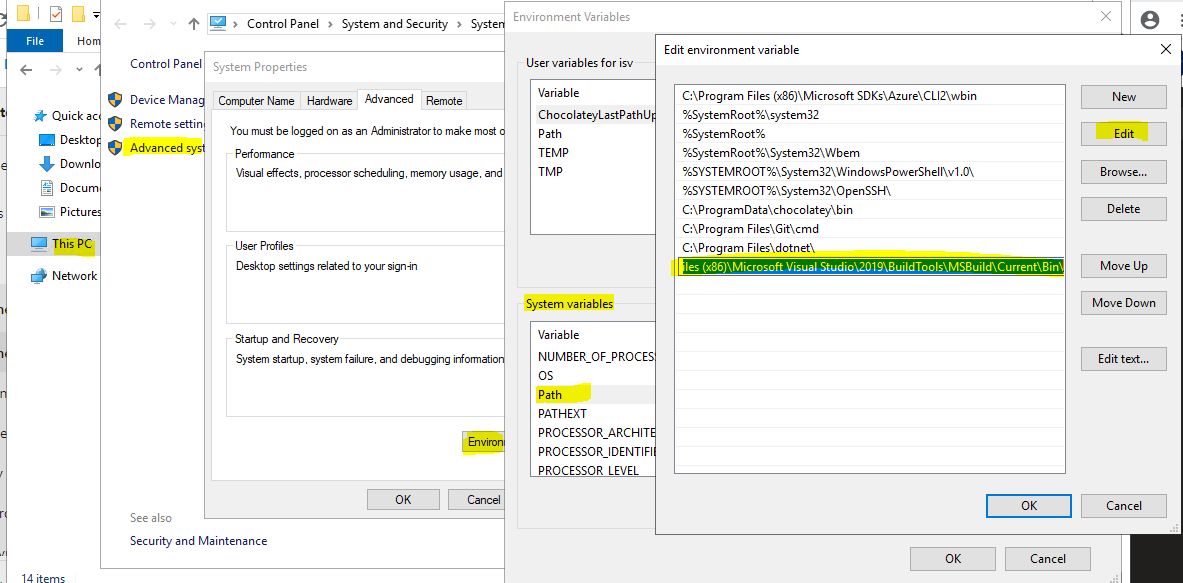
* Once restarted, open the below path and check MSBuild.exe is avaiable or not

C:\Program Files (x86)\Microsoft Visual Studio\2019\BuildTools\MSBuild\Current\Bin\

* Copy the MSBuild.exe path and right click on ThisPC(my computer)-> Properties -> click on Advanced System Settings -> click on Environment variables



**->select Path under System variables -> click on Edit -> click on New -> Paste the copied MSBuild.exe path -> click on ok**



Once path set and Open PowerShell and run the below command and check below command output/parameter values

MSBuild.exe /?

\*Q) on what basis u will recognize which platform its running?

based on configuration properties will say its running on x64 or x86

* Need to install github in vm, run the below command in PowerShell

choco install git.install

* After install, type 'git' in PowerShell. sometimes getting error, to load need to restart the machine. Otherwise open cmd prompt as administrator and type 'git' and enter.
* Need to install **nuget** packages in vm, run the below command in PowerShell

choco install nuget.commandline

* After install, type 'nuget.exe' in powershell
* Need to install google chrome in vm, run the below command in powershell

choco install googlechrome

choco install notepadplusepluse

* Open googlechrome and search for 'asp.net mvc 5 projects in github' and select the project (ex:SmartStoreNet)
* **Click on Clone in SmartStoreNet project and copy the url** (https://github.com/smartstore/SmartStoreNET.git)
* Need to create new directory(folder) in your local Drive, run the below command in PowerShell

New-Item C:\GitHubProjects\Traditional -ItemType Directory

* Need to change the directory in cmd prompt to newly created directory, run the below command in cmd prompt

cd C:\GitHubProjects\Traditional

* Need to clone(get) the project files from github into the newly created directory, run the below command in cmd prompt

git clone <github url> ---🡪copying the project into local path

git clone https://github.com/smartstore/SmartStoreNET.git

* After cloning, check all the project files are downloaded in directory (C:\GitHubProjects\Traditional)
* Now need to restore the nuget packages of the project, run the below command in PowerShell

cd C:\GitHubProjects\Traditional\SmartStoreNET

nuget restore src\SmartStoreNET.sln

Note: nuget restore : will restore all project dependence reference/dll

**Build The project:** After restoring nuget successfull, need to build the project with buildengine msbuild ".proj/.sln", run the below command in powershell

msbuild src\SmartStoreNET.sln -t:build

Build is failed due to missing of **'microsoft.webapplication.target'** but packages are installed.

* **Open Visual Studio Installer and Install VS-2019 -> click on Modify ->**

select the below tools under WorkLoads

-C++ Build tools

-.NET desktop build tools

-Universal Windows Platoform build tools

-web development build tools

-Data storage and processing build tools

-Node.js build tools

-.NET core build tools

select the below tools under Individual Components

-select all frameworks except EOL under .Net

-select nuget package manager under Code Tools

-select windows workflow foundation build tools under development activities

click on modify

\*Q)Build(compliation) completed where is the output?

it will store in the bin of the same project but it will not accurate, we will set specifically the output folder

* After installation of Visual Studio 2019 and run the below command in PowerShell

msbuild src\SmartStoreNET.sln -t:build -p:configuration="Release" -p:platform="Any CPU" -p:outdir="C:/Test"

**Note : output(atrifacts) stored in Test directory,**

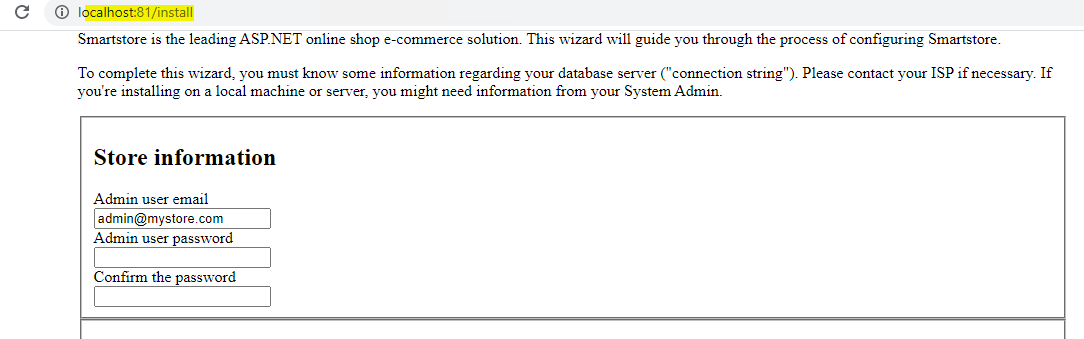
**Installing IIS:** For Testing purpose, whether build is happened correctly.

Install IIS in vm, execute the below command in PowerShell

Install-WindowsFeature -Name web-server -IncludeManagementTools

Install-WindowsFeature -Name web-server -IncludeAllSubFeature

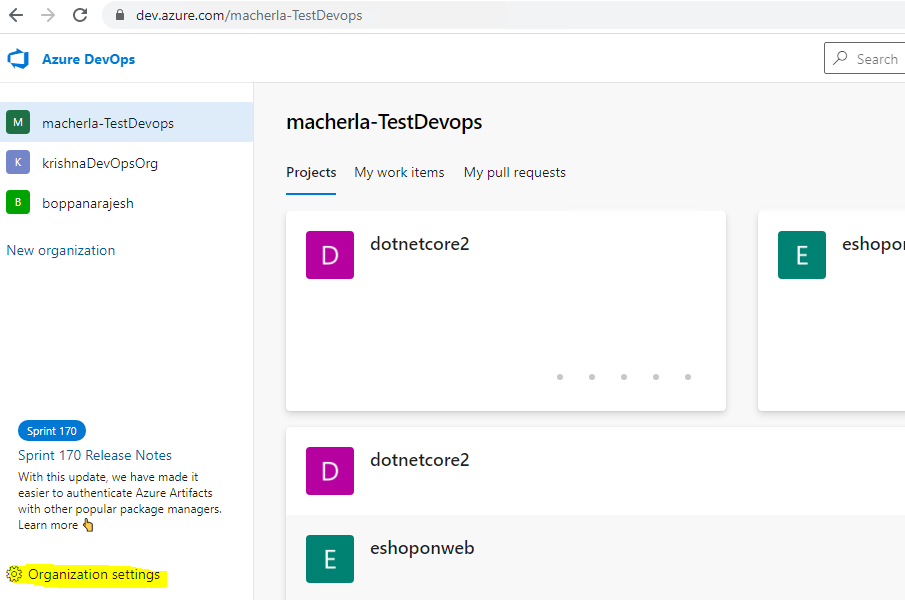
* open IIS(inetmgr), create new website and add output(Test) path(C:\Test\\_PublishedWebsites\SmartStore.Web) and run the site
* its throwing error "Could not find a part of the path *'C:\Test\\_PublishedWebsites\SmartStore.Web\bin\roslyn\csc.exe'."*
* copy the roslyn folder in proj folder and paste in bin of Test folder and re-run the website.(it working as excpected)



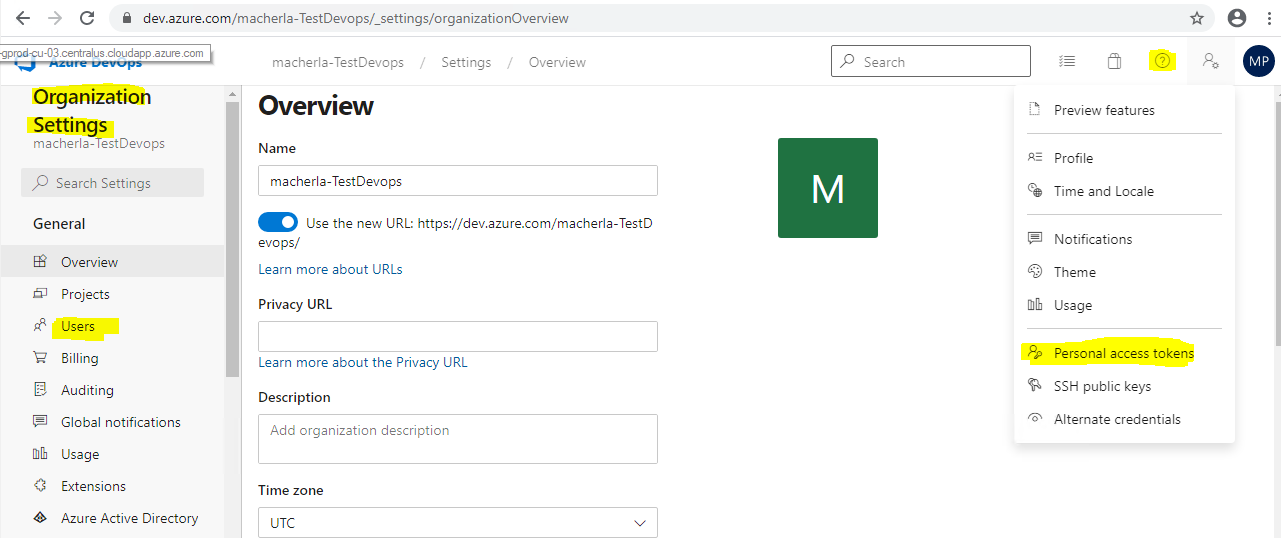
#####################################################################################

**AGent setp :**

* **Login into** [https://dev.azure.com/](https://dev.azure.com/macherla-TestDevops/dotnetcore2/_build/results?buildId=22&view=logs&j=275f1d19-1bd8-5591-b06b-07d489ea915a&t=78db2542-c627-4140-8a7a-d06178fff4e4)
* **Goto Organization setting**



**Creation of PAT (Personal Access Token) :** Top Right hand side select usersettings --> choose personal tokens -->Add token --> enter name , date time , choose full accesss--> save/add



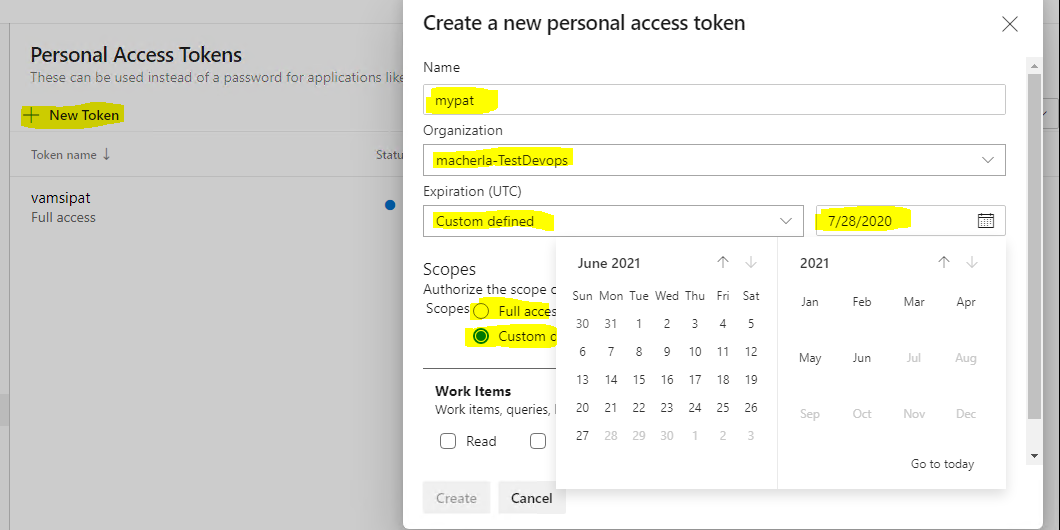
Select New Token->

Name: mypat

Organization:

Expirations : select custom and we can set it to up to one year

Scopes : Full access



* **IMP** : Copy the token code before closing the windows ( we can't retrieve once closed. so careful)

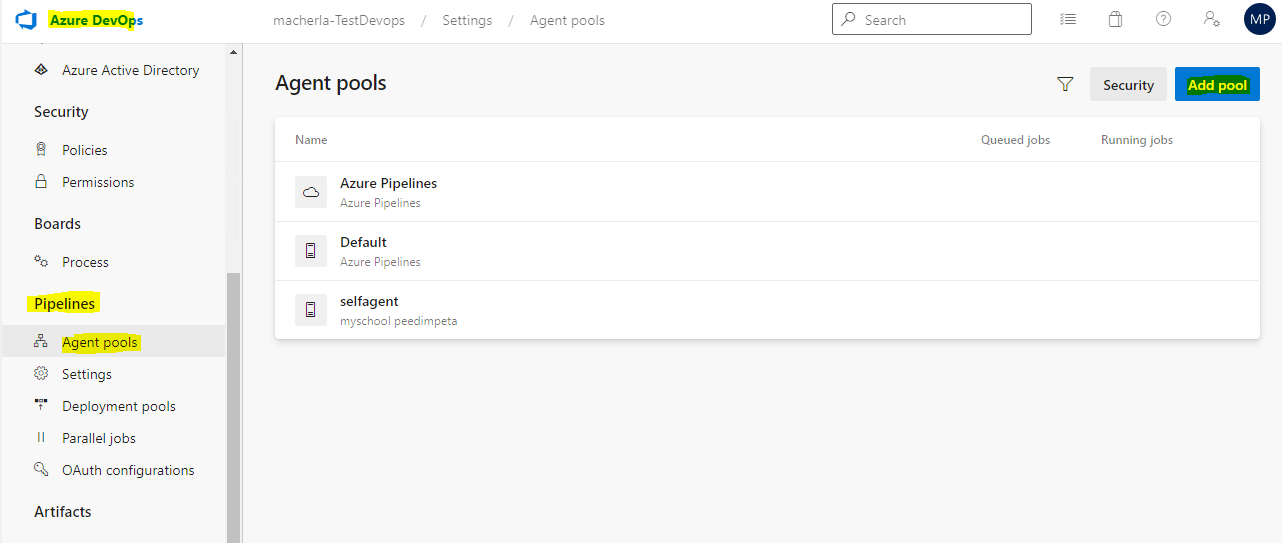
**Agent setup example:**

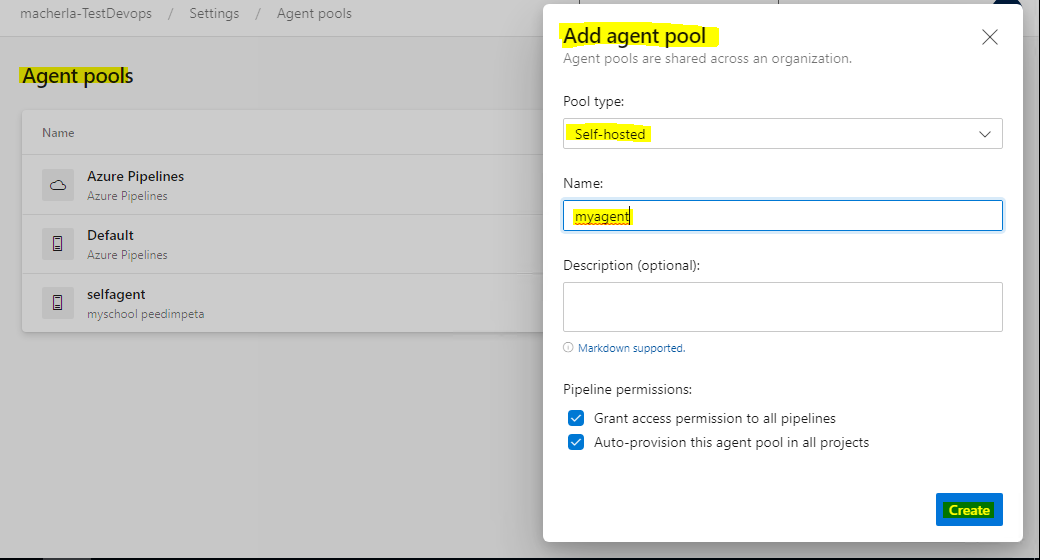
Under Organization Settings ->Expand Pipelines menu ->select Agent Pools-> Select “Add pool” in left side corner **-> select self hosted (from dropdown)**

**Pool Type : Self hosted**

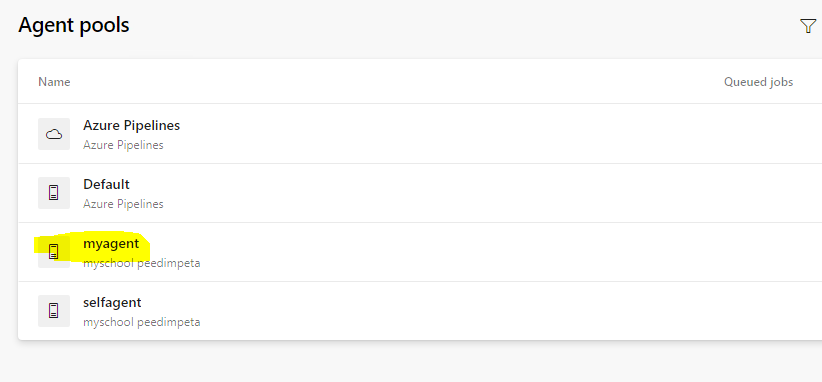
**Name : MyAgent**

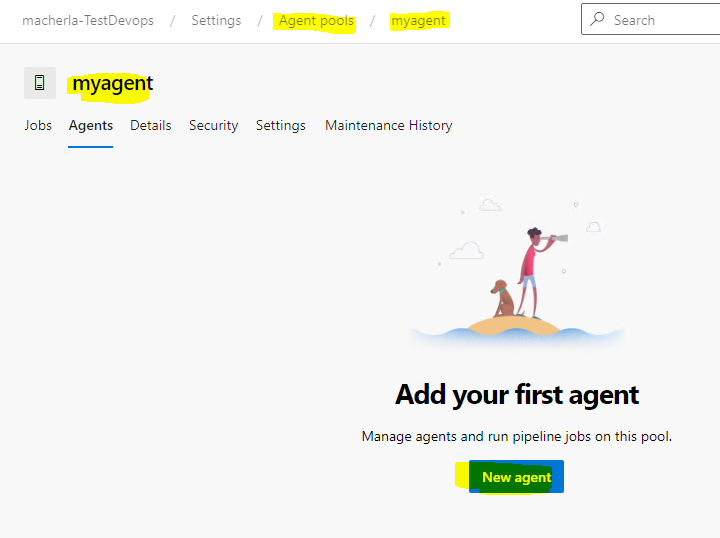
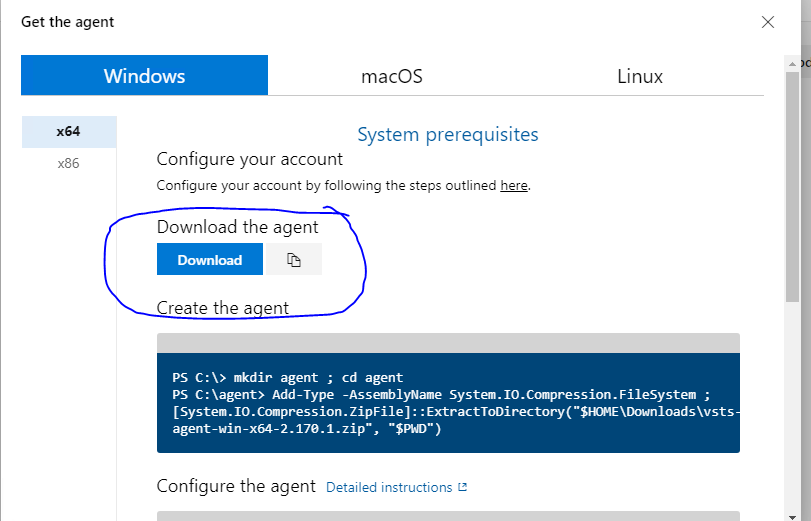
**Click on Create button**



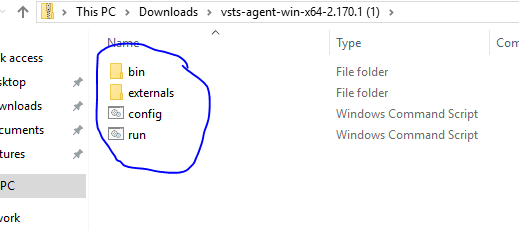


Once Agent created and download the Agent in your VM and install in your VM via PowerShell script like below.



Note : Download the agent -> unzip the file -> copy agent files into your local drive folder (C:\AZAgent)

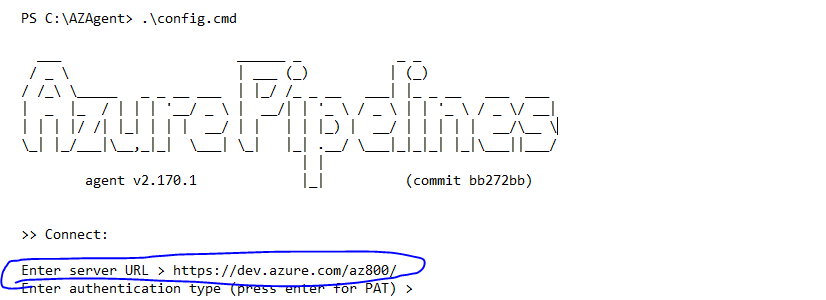


* Open PowerShell in administrator mode and change the directory to agent path

Cd C:\AZAgent

* Run below file

PS C:\AZAgent> .\config.cmd



* Past/Type PAT (personal access token key) key here

Enter personal access token > \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Connecting to server ...

>> Register Agent:

**Note: Enter valid PAT key then only Agent registered successfully otherwise it fails**

* Enter Your Agent Name

Enter agent pool (press enter for default) > myagent

Enter agent name (press enter for Vm1) > myagent -Agent1

Scanning for tool capabilities.

Connecting to the server.

Successfully added the agent

Testing agent connection.

Enter work folder (press enter for \_work) > C:\Agent\_workSpace (user defined name & one folder will create here with this name)

2020-06-25 03:07:10Z: Settings Saved.

Enter run agent as service? (Y/N) (press enter for N) > Y

* **Please enter your VM credentials**

Enter User account to use for the service (press enter for NT AUTHORITY\NETWORK SERVICE) > **isv (username)**

Enter Password for the account Vm1\azrajesh > **\*\*\*\*\*\*\*\*\*\*\*\* (password)**

Error reported in diagnostic logs. Please examine the log for more details.

- C:\AZAgent\\_diag\Agent\_20200625-030436-utc.log

Granting file permissions to 'Vm1\azrajesh'.

Service vstsagent.az800.AZ800.Az800-Agent1 successfully installed

Service vstsagent.az800.AZ800.Az800-Agent1 successfully set recovery option

Service vstsagent.az800.AZ800.Az800-Agent1 successfully set to delayed auto start

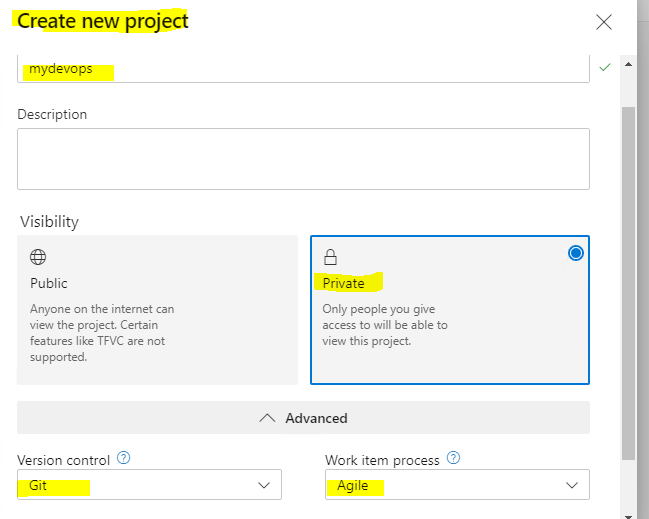
Service vstsagent.az800.AZ800.Az800-Agent1 successfully configured

Service vstsagent.az800.AZ800.Az800-Agent1 started successfully

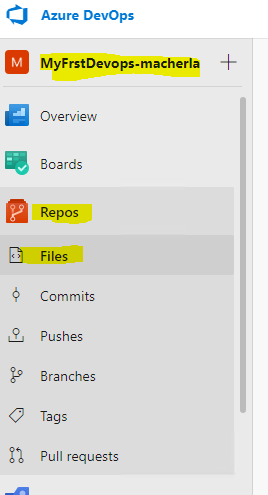
PS C:\AZAgent>

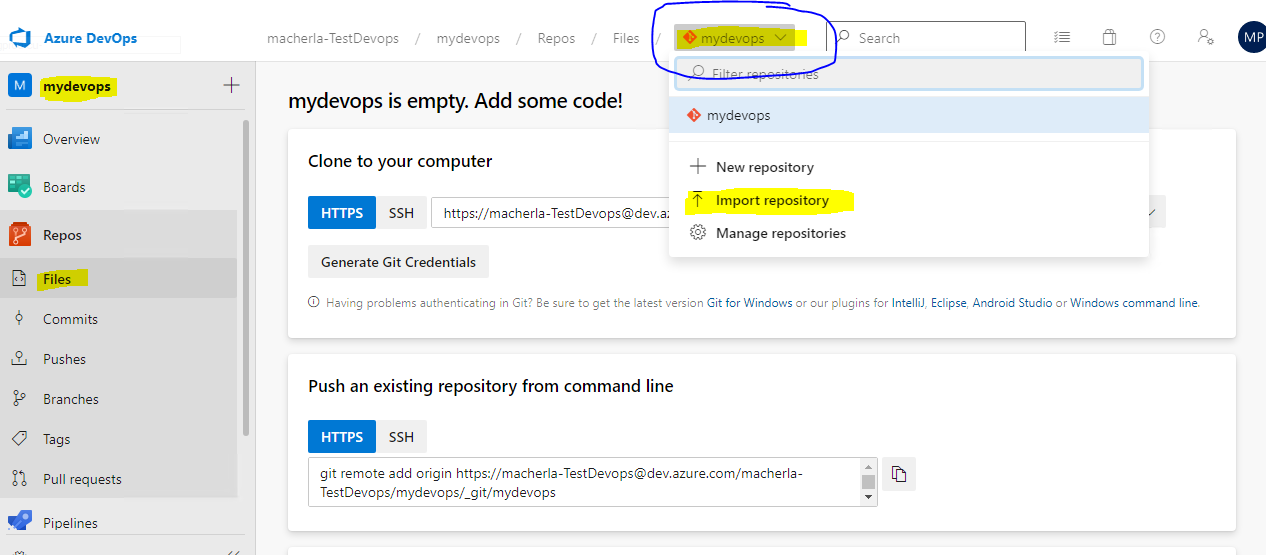
**Clone the GIT project the Devops portal , build and Generate the ARTIFACT**

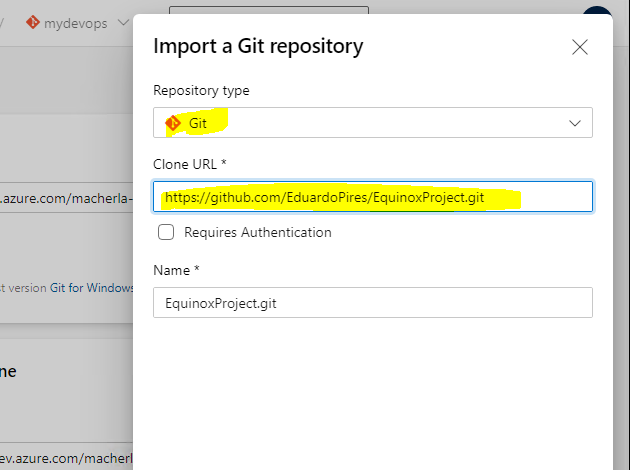
* Login into : [https://dev.azure.com/](https://dev.azure.com/macherla-TestDevops)
* Create Organization and under the create new project



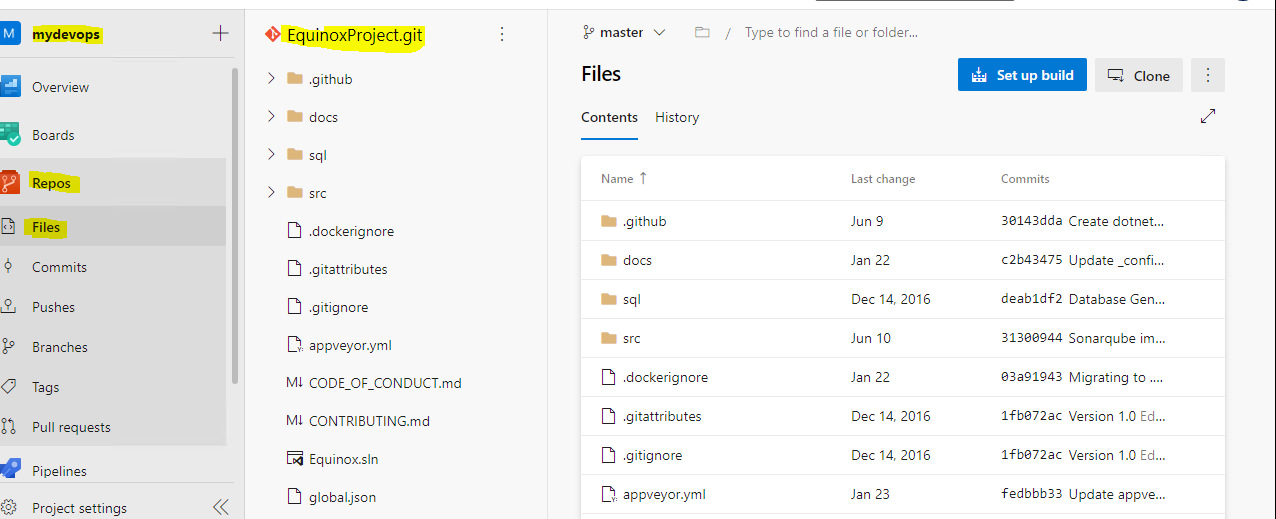
* Project will create and open this project ->select “Repos” -> Files



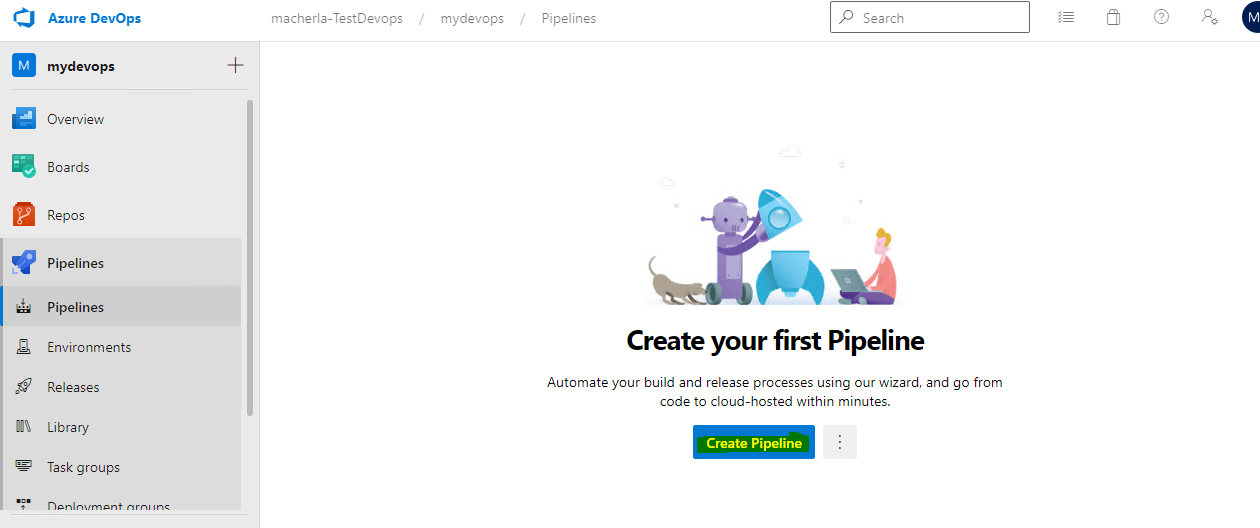


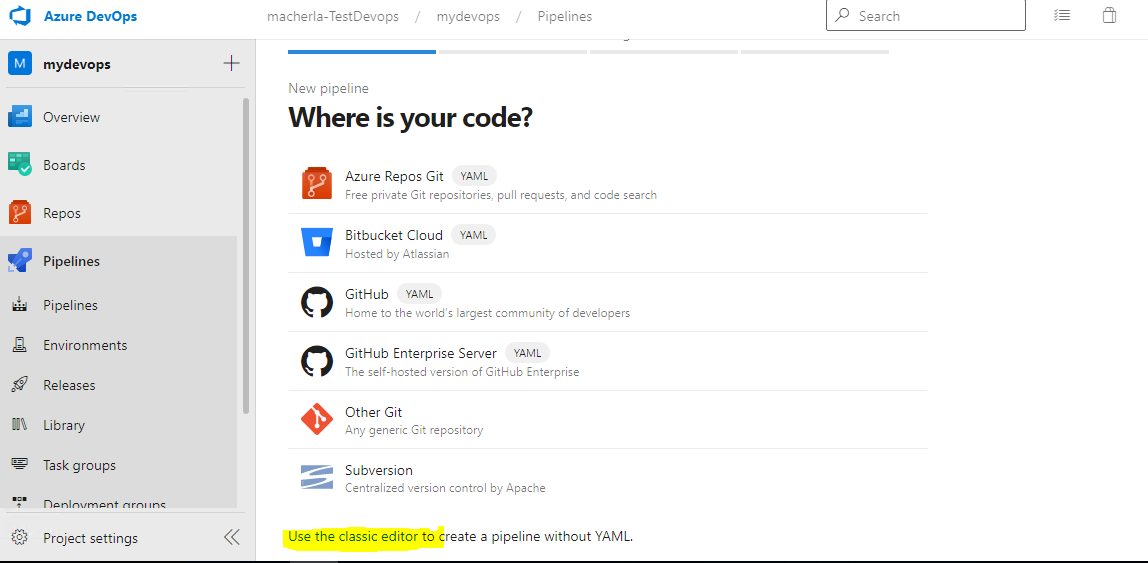
**Past GIT hub project URL into Clone URL text box**

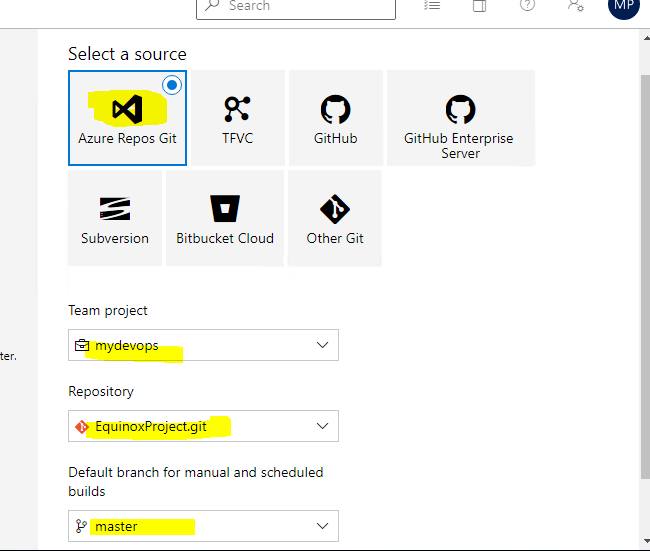
After few mins all your project file copied into your Devops project-> Repos-> files section

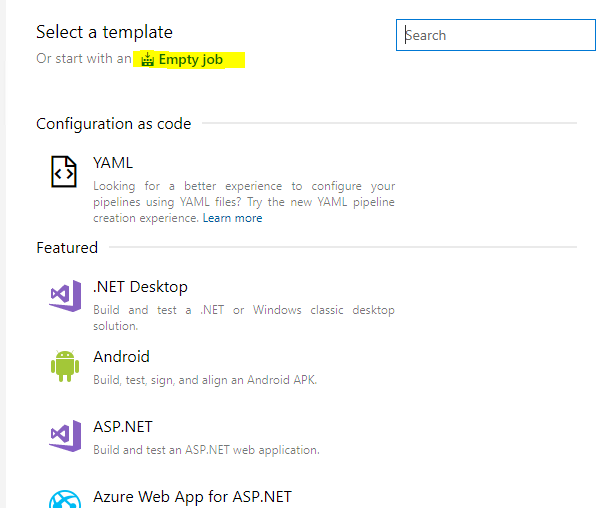


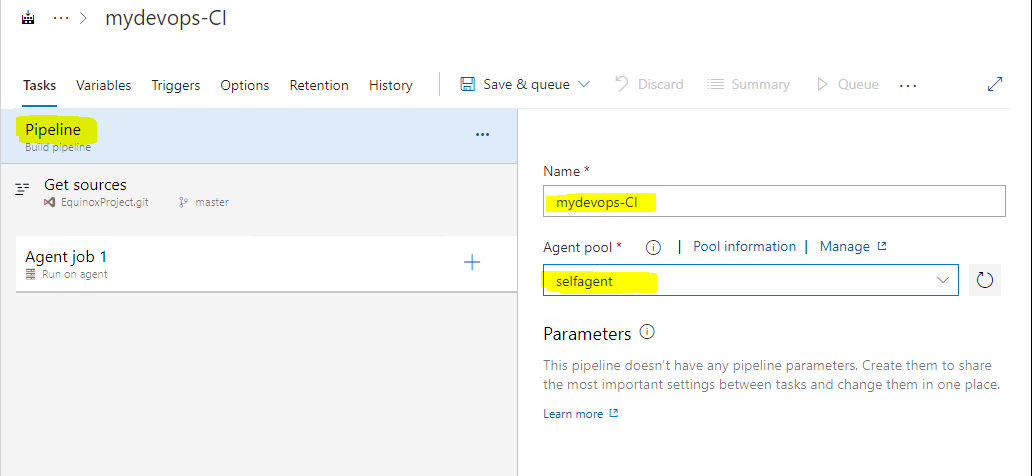
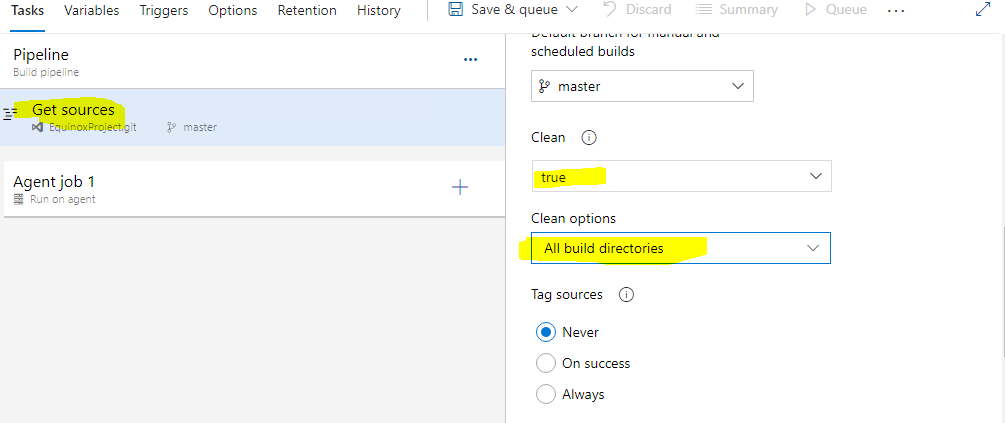
**Now we are creating Pipe Line for build the project :**



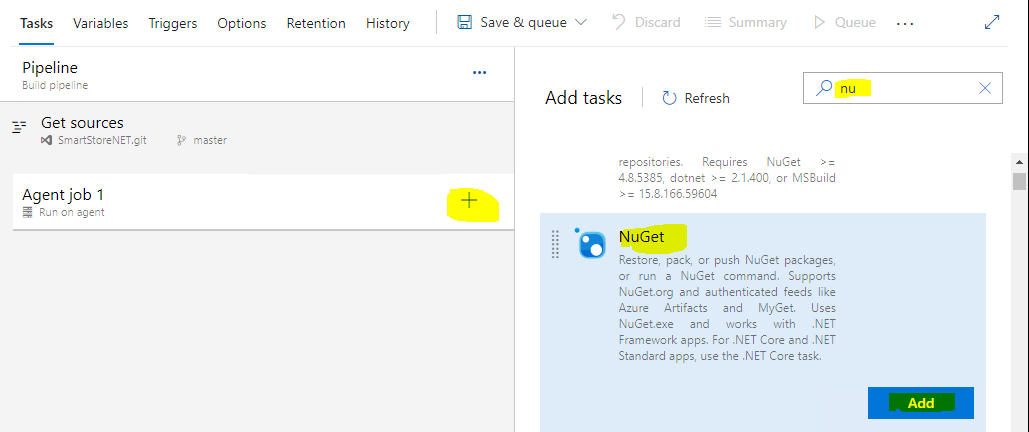






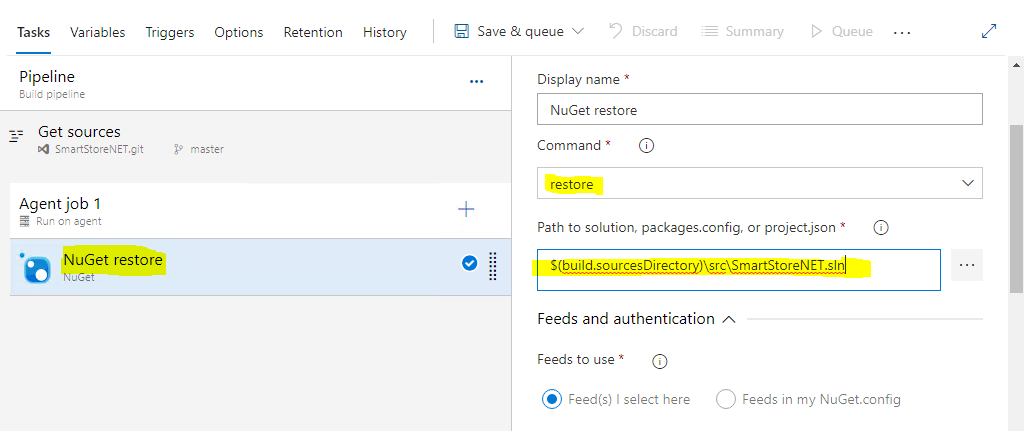
For Restoreing nuget pack refece add the Agent job like below:



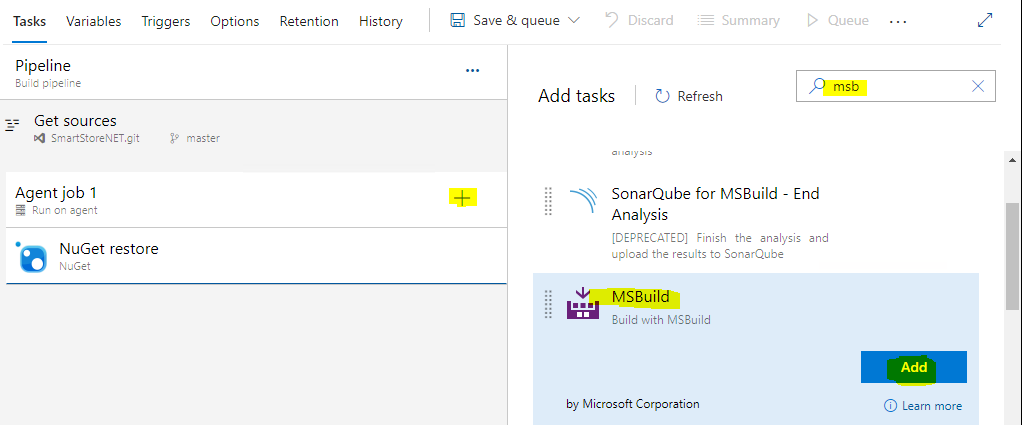
Name: Nuget restore

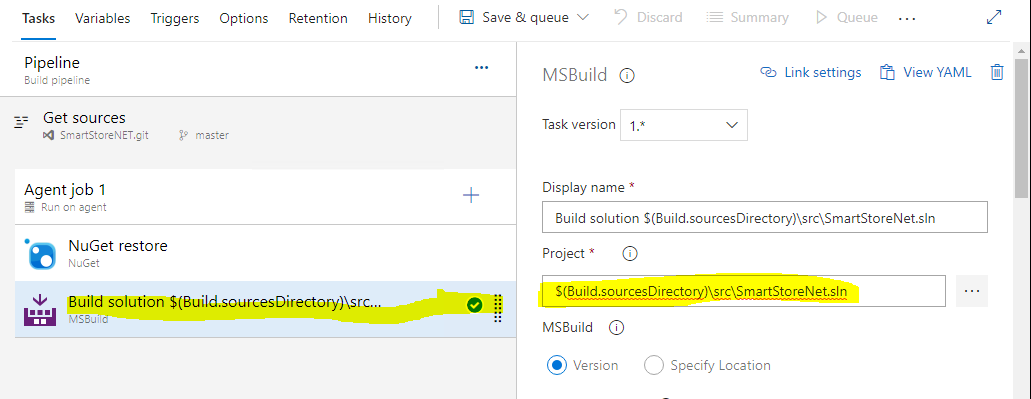
Command: restore

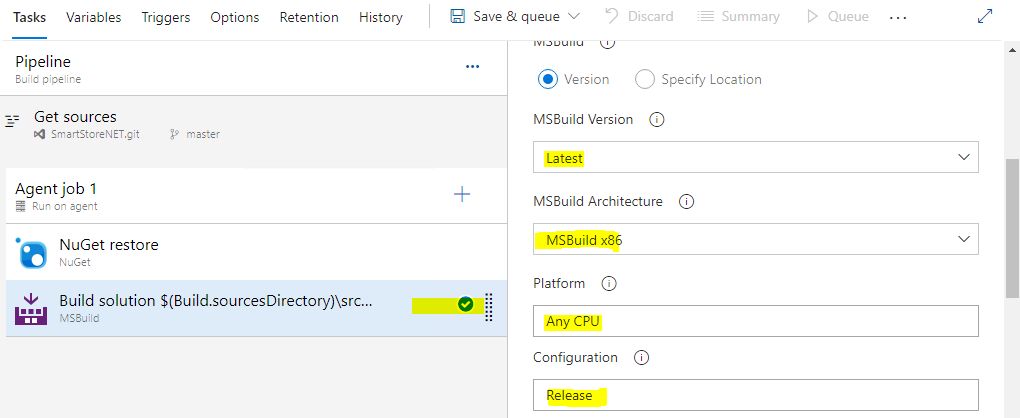
Path: $(Build.sourcesDirectory)\src\SmartStoreNet.sln



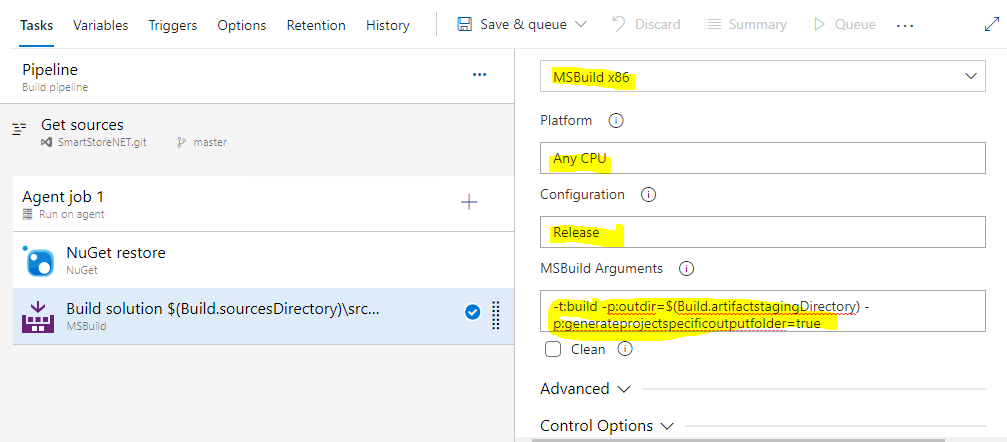
**Now Build the project by using MSBuild.exe by adding new agent job:**

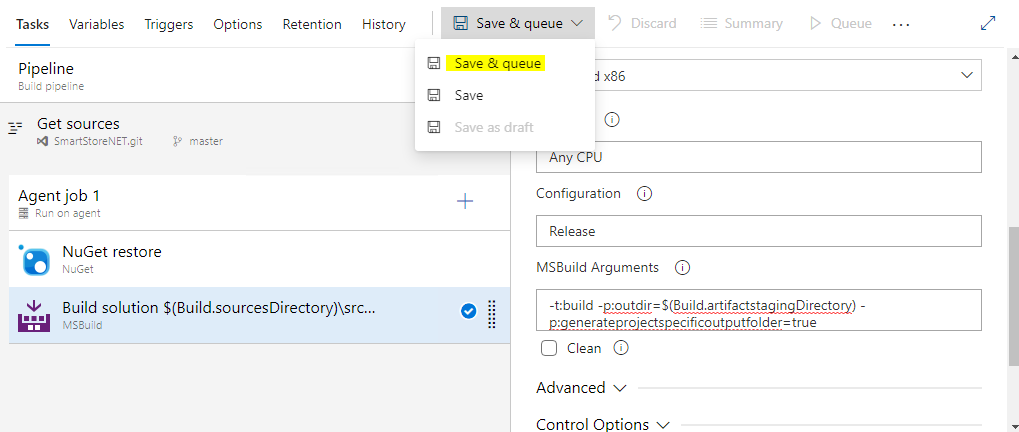


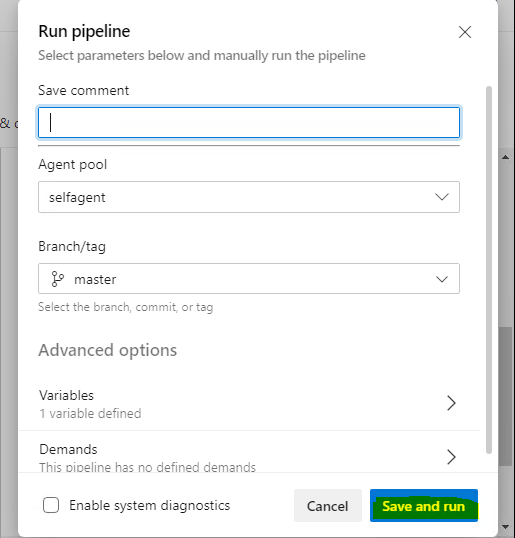




**MSBuild Arguments : -t:build -p:outdir=$(Build.artifactstagingDirectory) -p:generateprojectspecificoutputfolder=true**





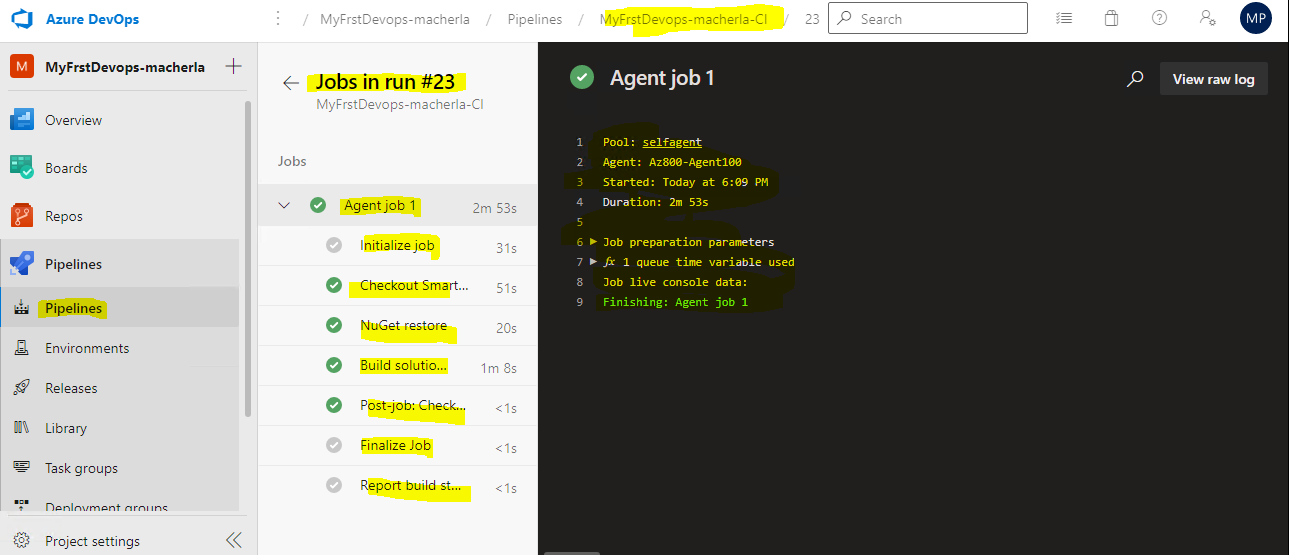


Ediing PipeLine:

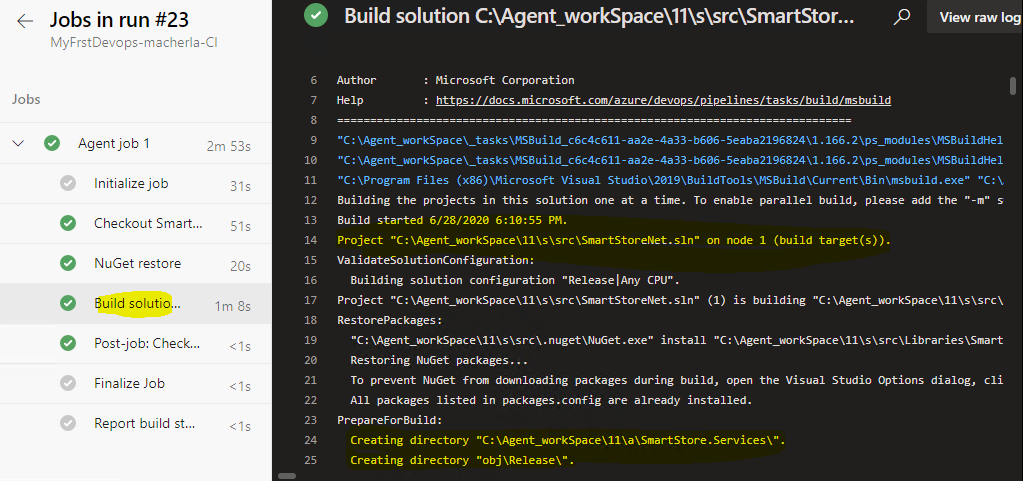
-t:build -p:outdir=$(Build.artifactstagingDirectory) -p:generateprojectspecificoutputfolder=true

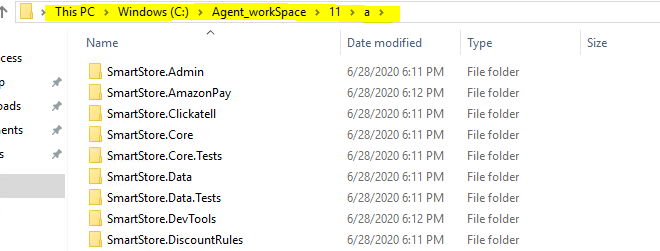
in option tab --> build number format --> $(Date:yyyyMMdd).$(Rev:r)-$(Build.DefinitionName)

Note: Once run your **PipeLine** your build got success and here you find build failure reasons as well

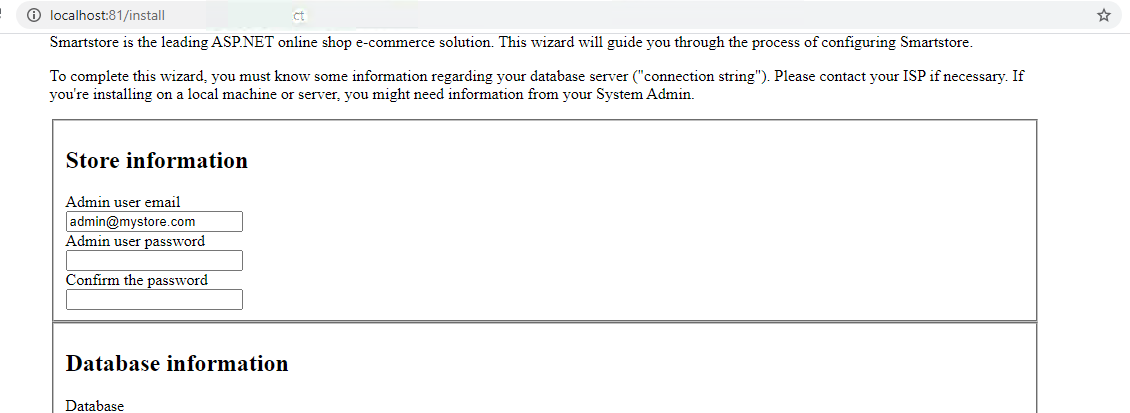


Your Artifact package files will store in your local drive work space folder

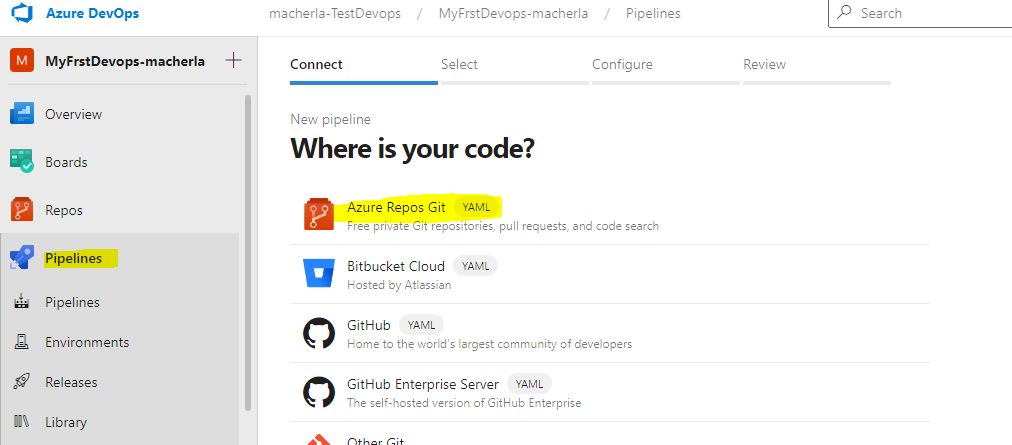




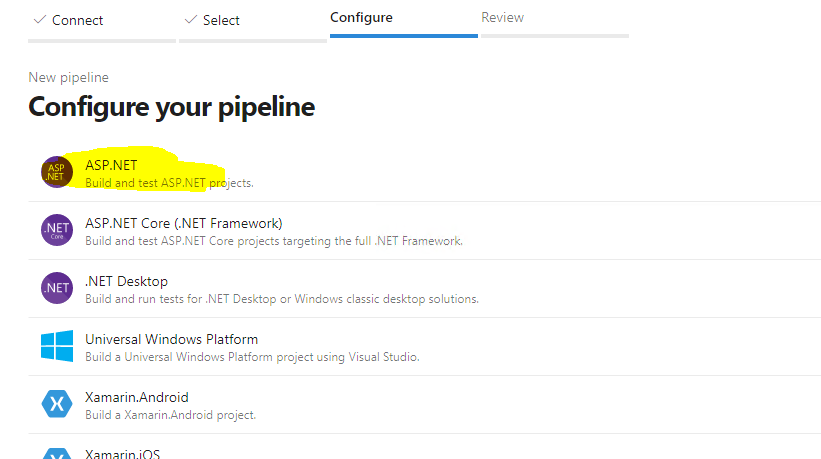
**You can use the Artifact zips file for deploy the site in IIS server**

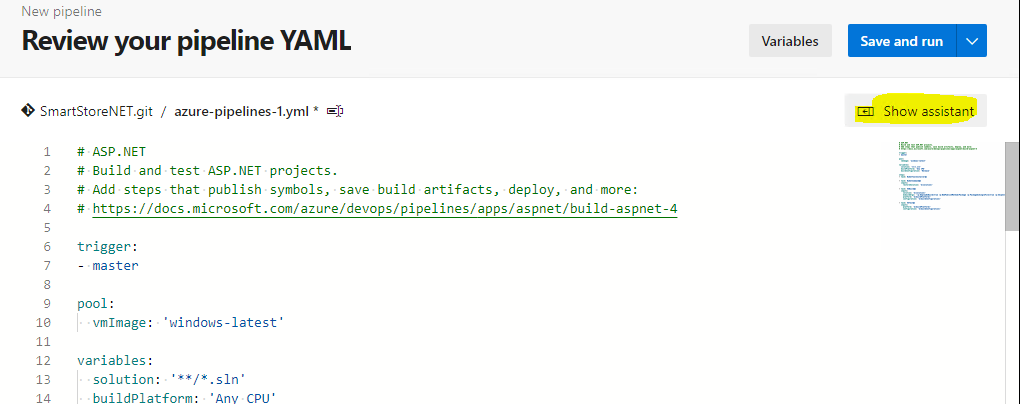


Build The project by using YML :

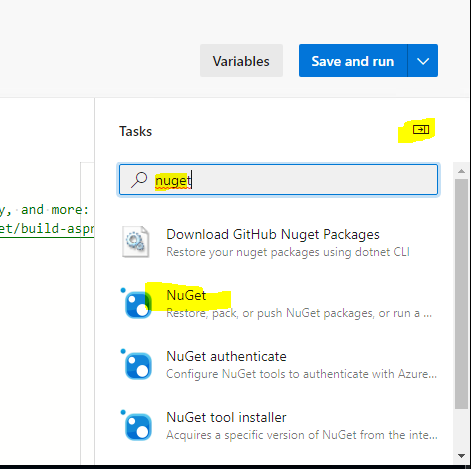
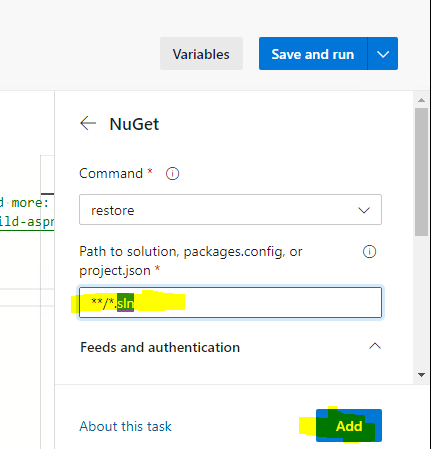


Select your Project and next

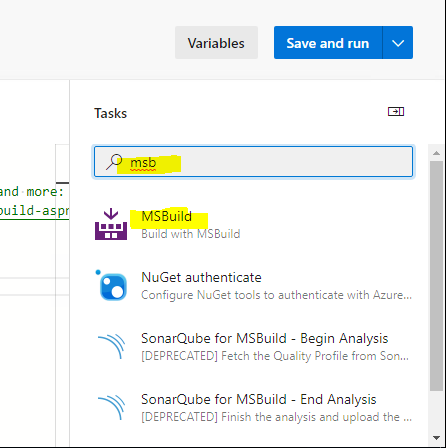


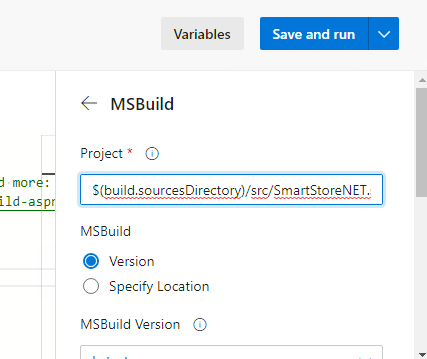


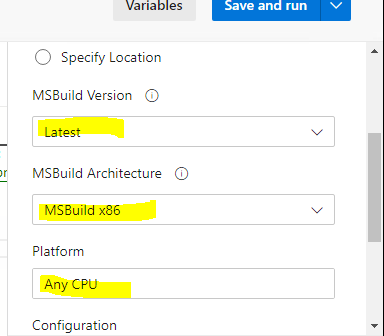
Note : Remove existing script and add your own script here by using “**show assistant”**

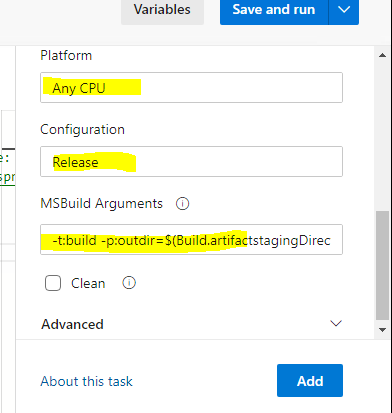
 

**Add MSBuild Task :**









**Note : Your whole YML file script like below**

# ASP.NET

# Build and test ASP.NET projects.

# Add steps that publish symbols, save build artifacts, deploy, and more:

# https://docs.microsoft.com/azure/devops/pipelines/apps/aspnet/build-aspnet-4

trigger:

- master

pool:

  name: 'selfagent'

variables:

  solution: '\*\*/\*.sln'

  buildPlatform: 'Any CPU'

  buildConfiguration: 'Release'

steps:

- task: NuGetCommand@2

  inputs:

    command: 'restore'

    restoreSolution: '\*\*/\*.sln'

    feedsToUse: 'select'

    vstsFeed: '72c5dc0d-cb8d-4805-bce7-7225157f2539'

- task: MSBuild@1

  inputs:

    solution: '$(build.sourcesDirectory)/src/SmartStoreNET.sln'

    platform: 'Any CPU'

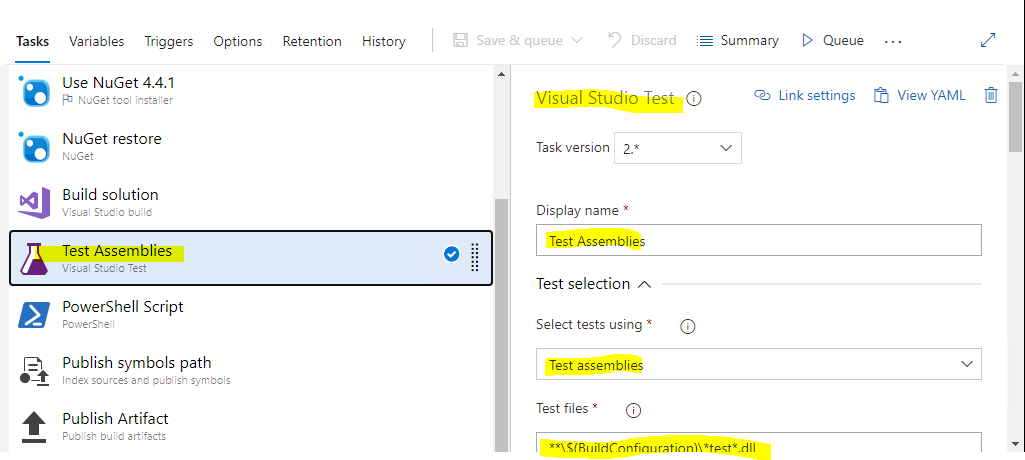
    configuration: 'Release'

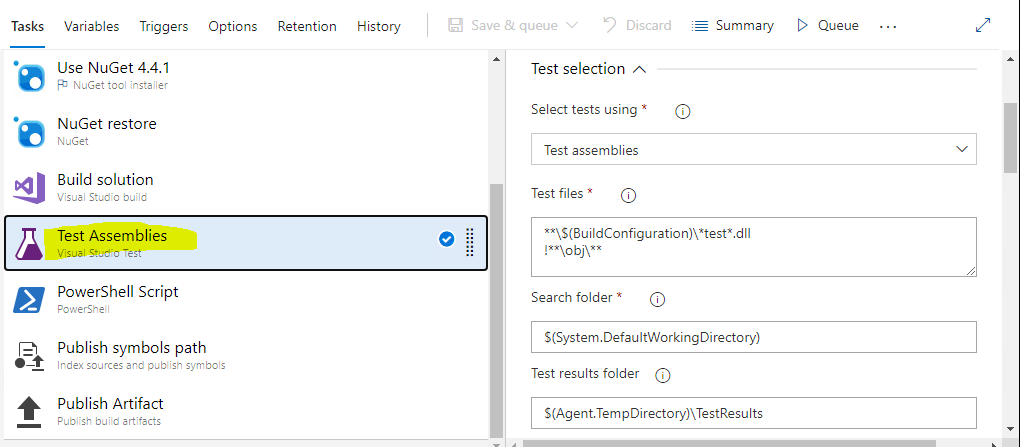
    msbuildArguments: '-t:build -p:outdir=$(Build.artifactstagingDirectory) -p:generateprojectspecificoutputfolder=true'

Latter RUN the YML Build and these build files will save in local workspace folder

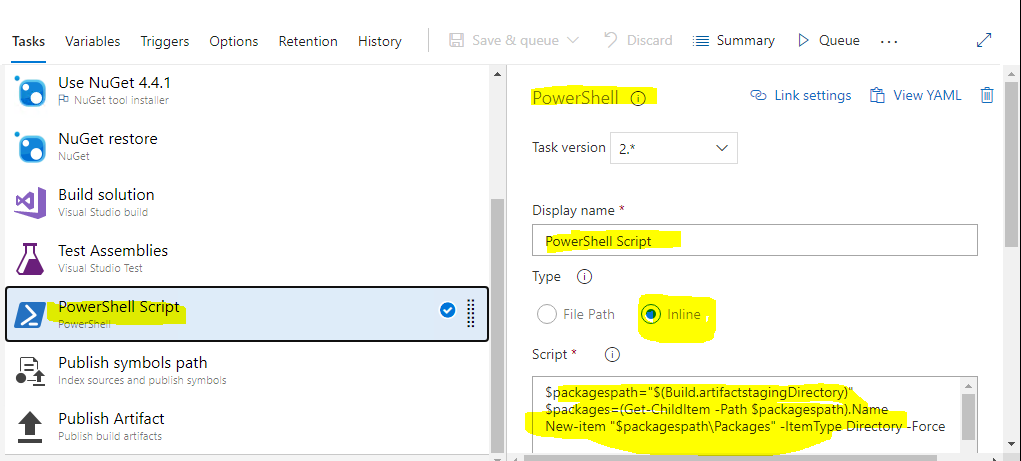
**Build and Generating ARTIFACT and Packaging by using PowerShell:**

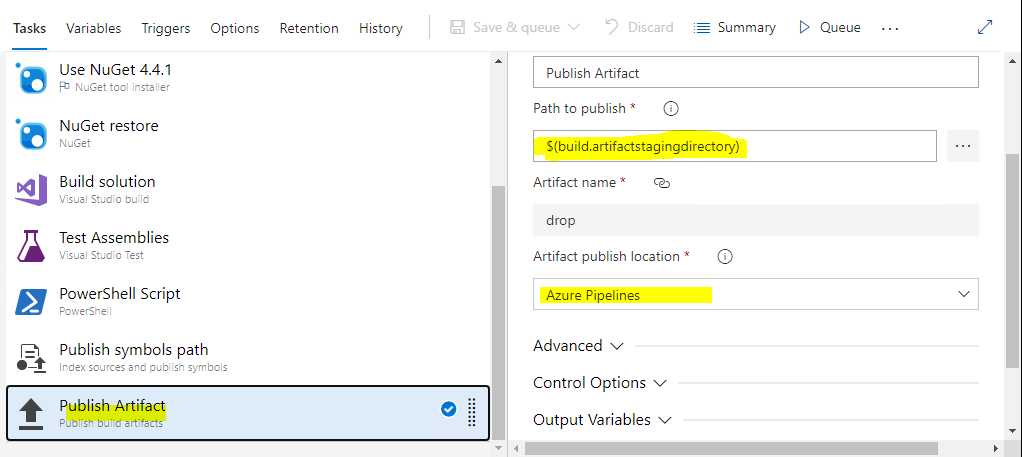
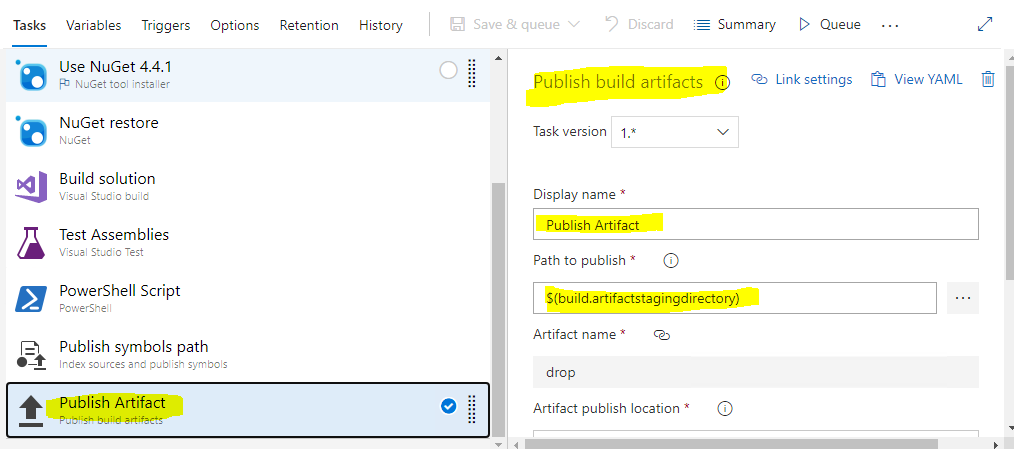
**Test Cases :**





**Powershell Task :**





######################## **PackagingArtifact task** ############################

Publishin PipelineArtFacts :

$(build.artifactstagingdirectory)\packages

############################ **PowerShell Script task Inline code** ############################

$packagespath="$(Build.artifactstagingDirectory)"

$packages=(Get-ChildItem -Path $packagespath).Name

New-item "$packagespath\Packages" -ItemType Directory -Force

if($packages.Length -ne 0){

Foreach($package in $packages){

if(Test-path "$packagespath\$package\\_Publishedwebsites")

{

$folderpath="$packagespath\$package\\_Publishedwebsites\$package\\*"

#zip the packages

Compress-Archive $folderpath -DestinationPath "$packagespath\packages\$package.zip" -Force -Verbose

}

else{

Write-Warning -Message "No published websites found in current package :$package folder "

}

}

}

else{

Write-Error -Message "No packages found in the current folder" -ErrorAction stop

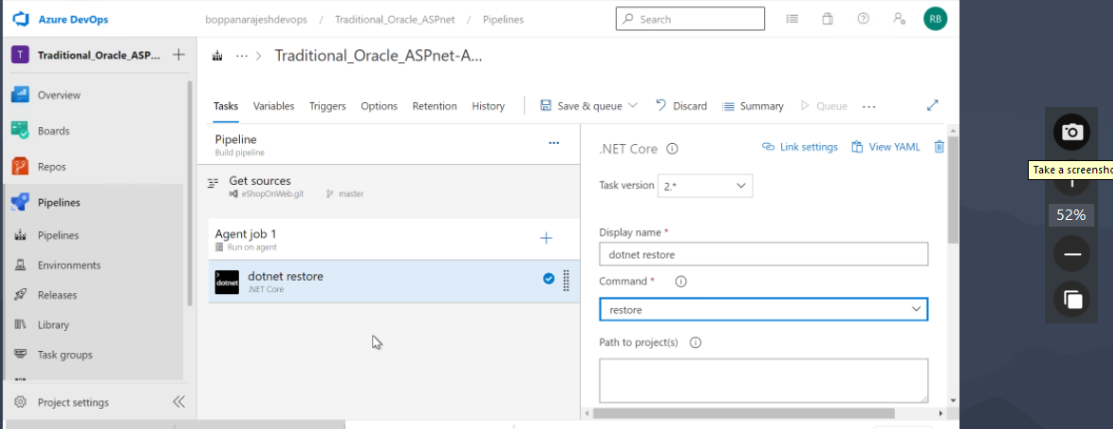
exit 1

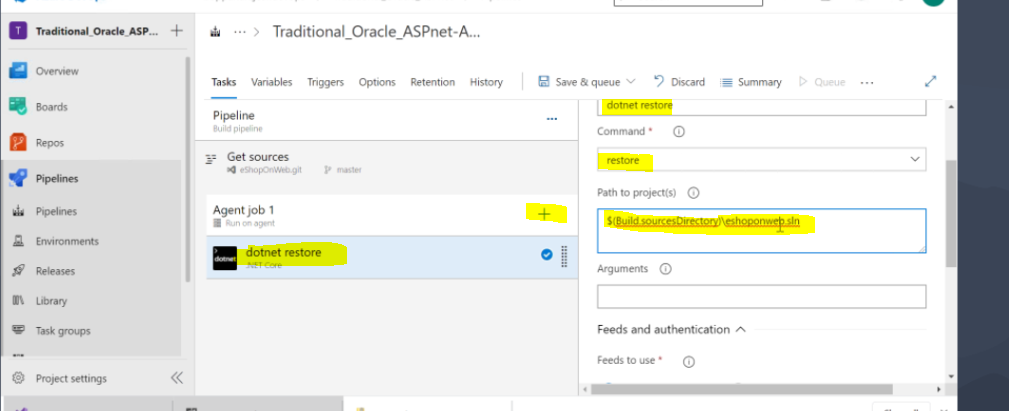
}

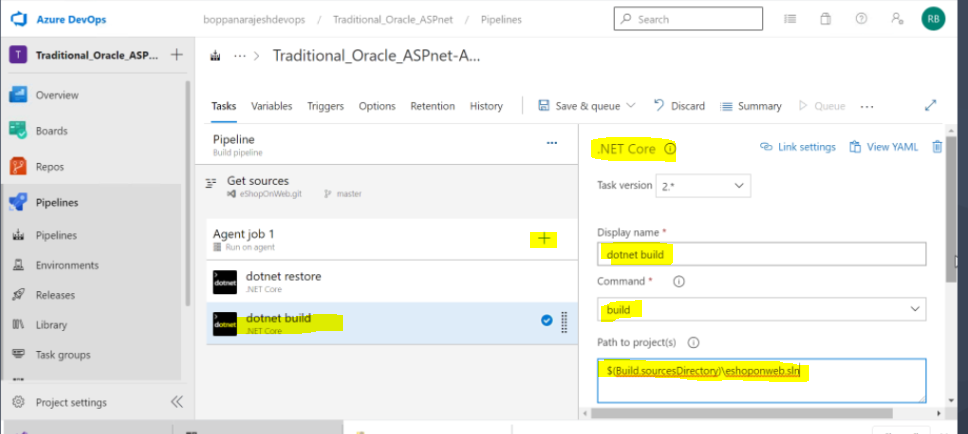
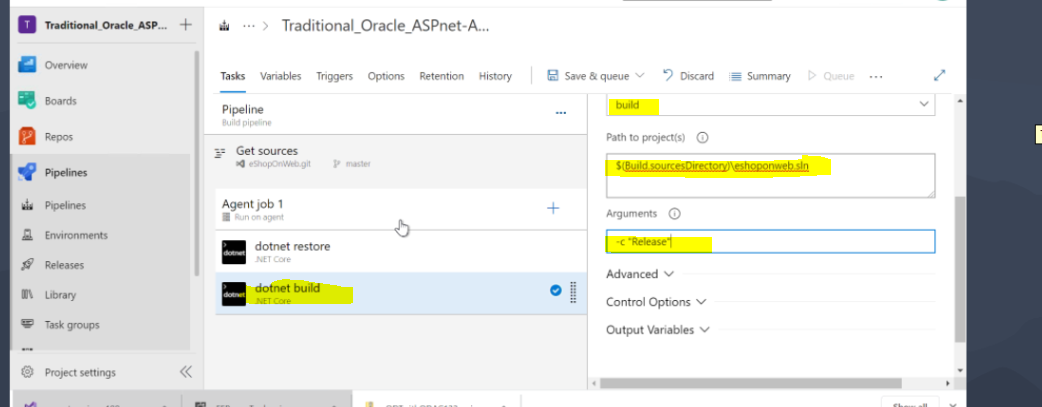
**DotNet Core Project Build Process:**

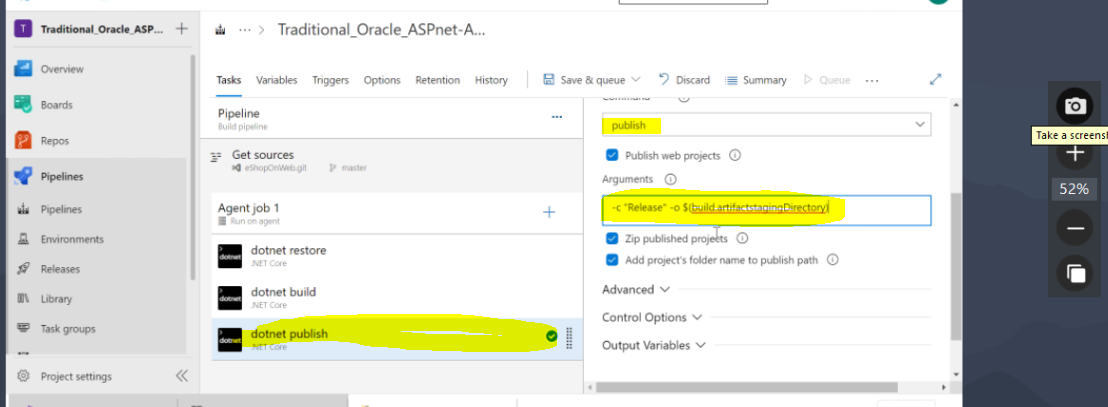
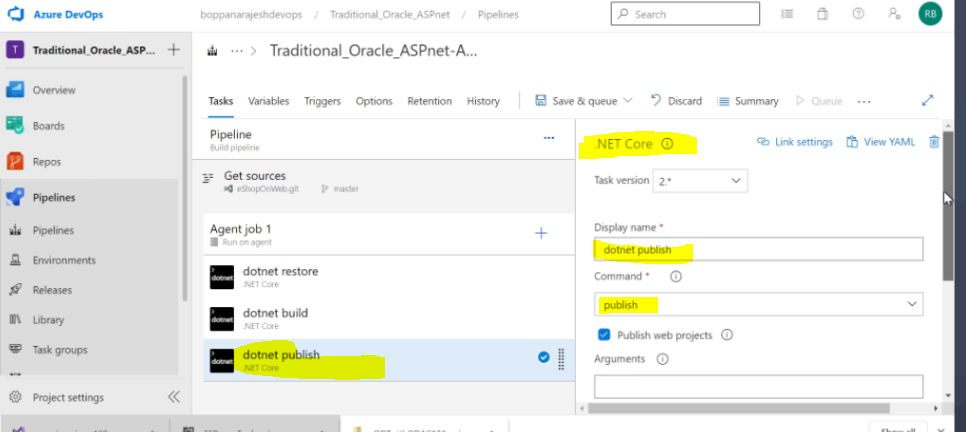
Variable values

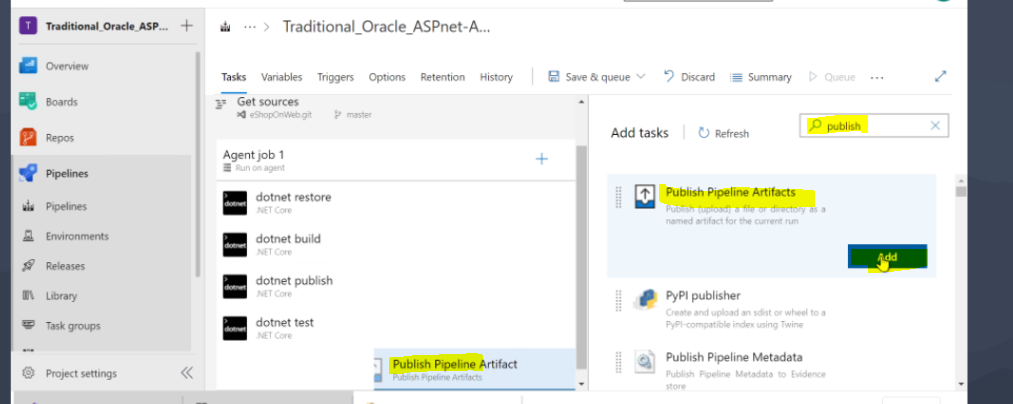
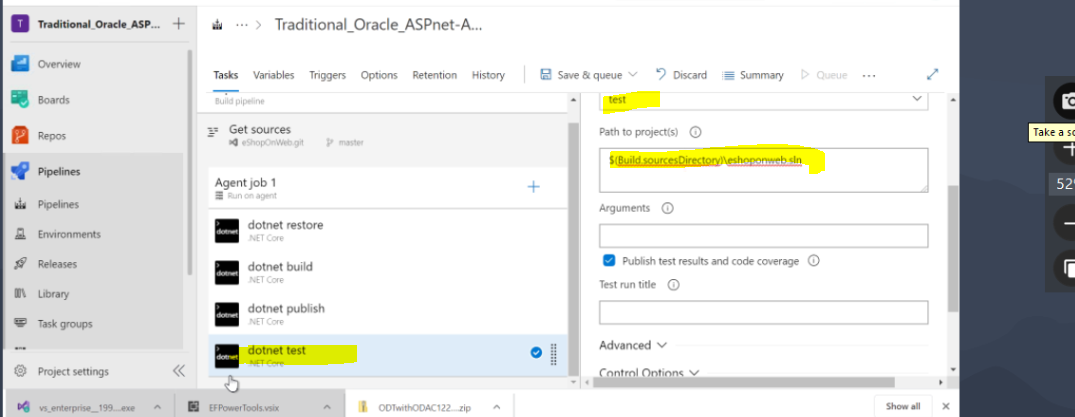


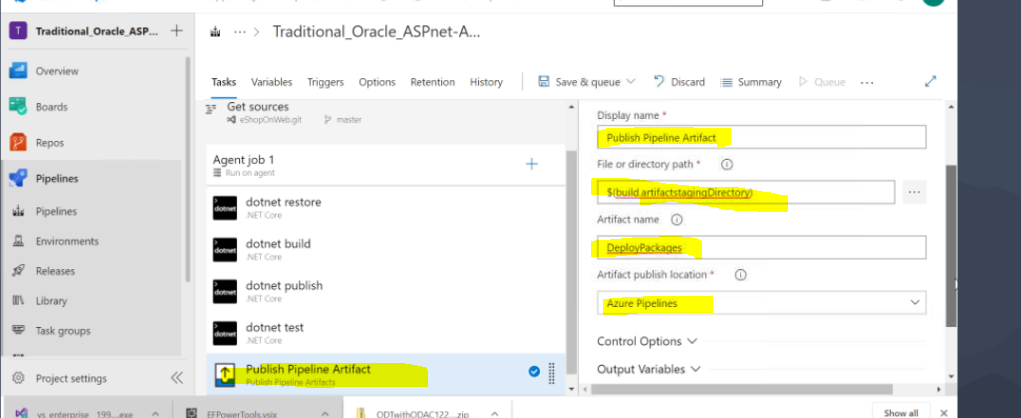
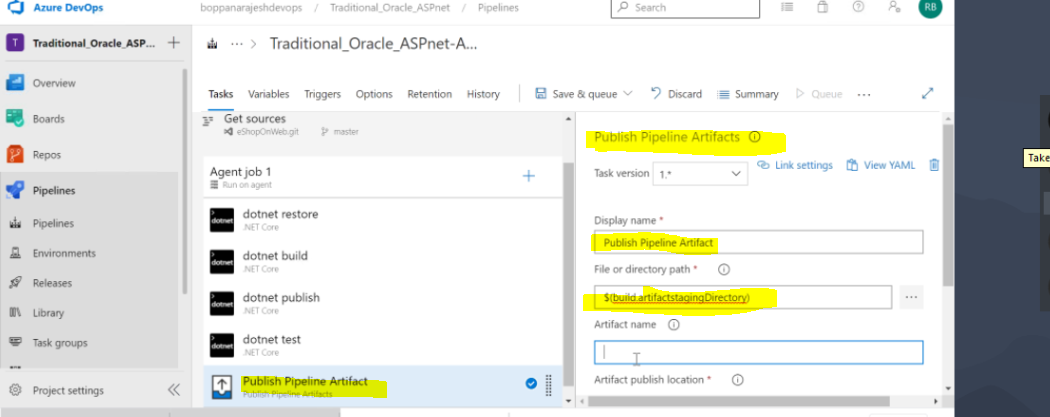




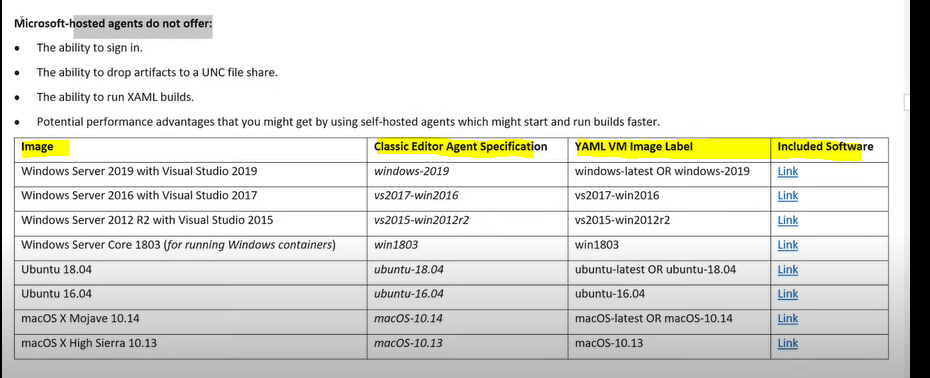


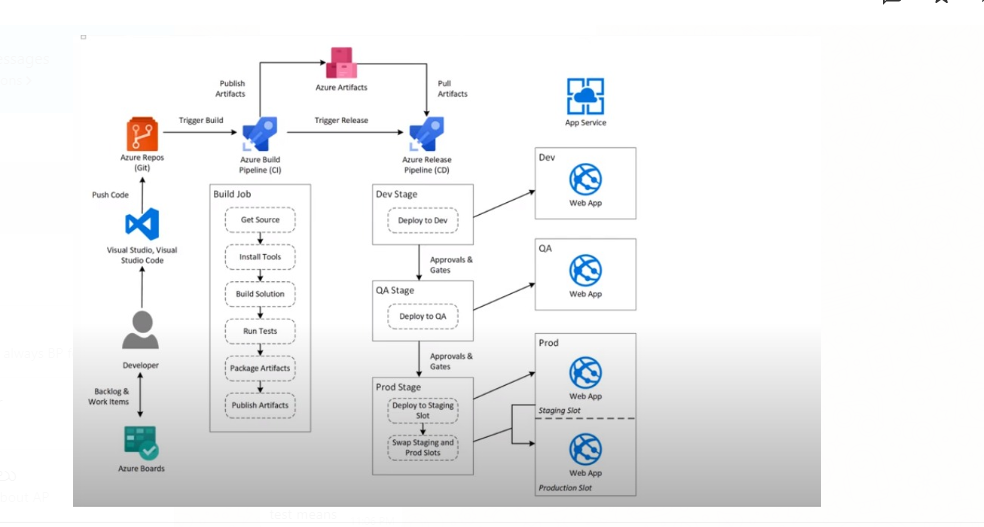


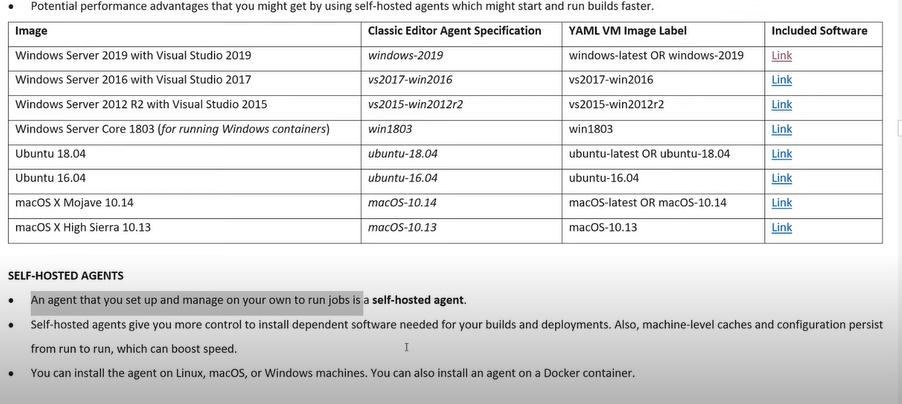




**Whole CI&CD process :**







#service principal

$user="xxxxxxxxxxxxxxx" # provide application\_id as user id

$secret=ConvertTo-SecureString -String "xxxxxxxxxxxxxxx" -AsPlainText -Force # provide password as client secret

$spcreds=New-Object -TypeName System.Management.Automation.PSCredential -ArgumentList ($user, $secret)

$tenant="xxxxxxxxxxxxxxx" # provide tenant ID

Connect-AzAccount -ServicePrincipal -Credential $spcreds -Tenant $tenant

# variables

$resourcegroup="RG360"

$location= "eastus"

$vnetname="ubantu-vnet"

$address\_space="10.49.0.0/16"

$subnet="ubantusubnet"

$address\_prefix="10.49.1.0/24"

$vmName="ubantuvm"

$vm\_admin\_user="azadmin"

#user credentials

$password=ConvertTo-SecureString ' ' -AsPlainText -Force

$creds=New-Object -TypeName System.Management.Automation.PSCredential -ArgumentList ($vm\_admin\_user, $password)

# resource

$rg=New-AzResourceGroup -Name $resourcegroup -Location $location

#subnet

$subnetconfig=New-AzVirtualNetworkSubnetConfig -Name $subnet -AddressPrefix $address\_prefix

#vnet

$vnet=new-azvirtualNetwork -Name $vnetname -ResourceGroupName $rg.ResourceGroupName -Location $rg.Location -AddressPrefix $address\_space -Subnet $subnetconfig

#public IP

$pip=New-AzPublicIpAddress -Name "$vmName-ip" -ResourceGroupName $rg.ResourceGroupName `

-Location $rg.Location -AllocationMethod Static

# create inbound rule to connect via ssh

$nsgSSHRule=New-AzNetworkSecurityRuleConfig -Name "sshrule" -Protocol Tcp -Direction Inbound `

-Priority 550 -SourceAddressPrefix \* -SourcePortRange \* -DestinationAddressPrefix \* -DestinationPortRange 22 -Access Allow

#create NSG

$nsg=New-AzNetworkSecurityGroup -ResourceGroupName $rg.ResourceGroupName `

-Location $rg.Location -Name "nsg630" -SecurityRules $nsgSSHRule

#nic

$nic= New-AzNetworkInterface -Name "$vmName-nic" -ResourceGroupName $rg.ResourceGroupName `

-Location $rg.Location -SubnetId $vnet.Subnets[0].Id -PublicIpAddressId $pip.Id -NetworkSecurityGroupId $nsg.Id

#create vm configuration

$vmconfig= New-AzVMConfig -VMName $vmName -VMSize Standard\_B2s | `

Set-AzVMOperatingSystem -Linux -ComputerName $vmName -Credential $creds -DisablePasswordAuthentication |`

Set-AzVMSourceImage -PublisherName Canonical -Offer UbuntuServer -Skus 18.04-LTS -Version latest| `

Add-AzVMNetworkInterface -Id $nic.Id

<#

choco install openssh

refreshenv

Get-Service -Name ssh-agent | Set-Service -StartupType Manual

start-service ssh-agent

get-serviec ssh-agent # status should be running mode

cd C:\users\your profile\.ssh

ssh-keygen.exe pres enter

again press enter key without supply any value

press enter key without supply key

Check for the files in your localpath C:\users\your profile\.ssh

ls

files will be listed like

Mode LastWriteTime Length Name

---- ------------- ------ ----

-a---- 6/28/2020 10:20 AM 1675 id\_rsa

-a---- 6/28/2020 10:20 AM 423 id\_rsa.pub

-a---- 9/16/2019 12:20 PM 191 known\_hosts

#>

#configure SSH key

$sshkey=Get-Content "$env:USERPROFILE\.ssh\id\_rsa.pub"

Add-AzVMSshPublicKey -VM $vmconfig -KeyData $sshkey -path "/home/$vm\_admin\_user/.ssh/authorized\_keys"

#create VM

New-azVM -ResourceGroupName $rg.ResourceGroupName -Location $rg.Location -VM $vmconfig