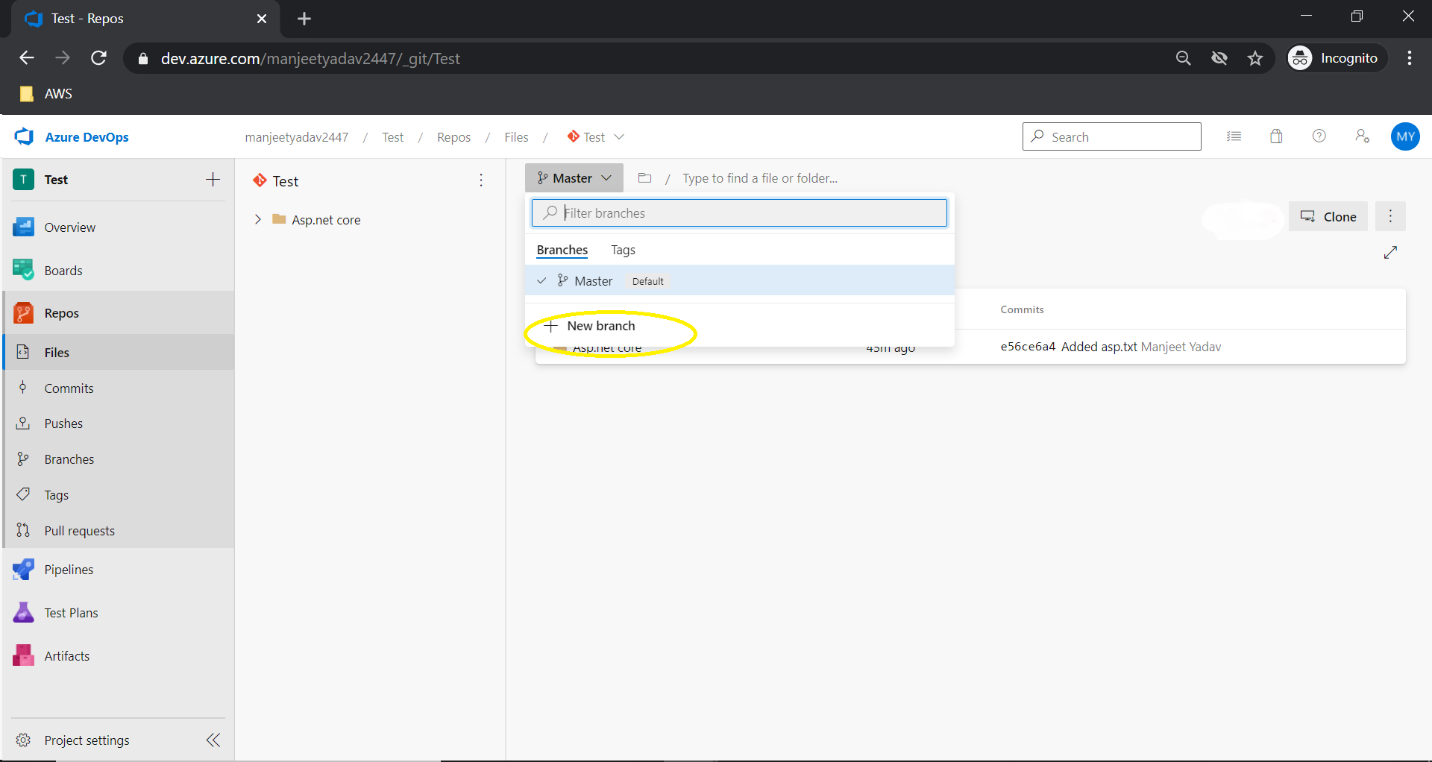
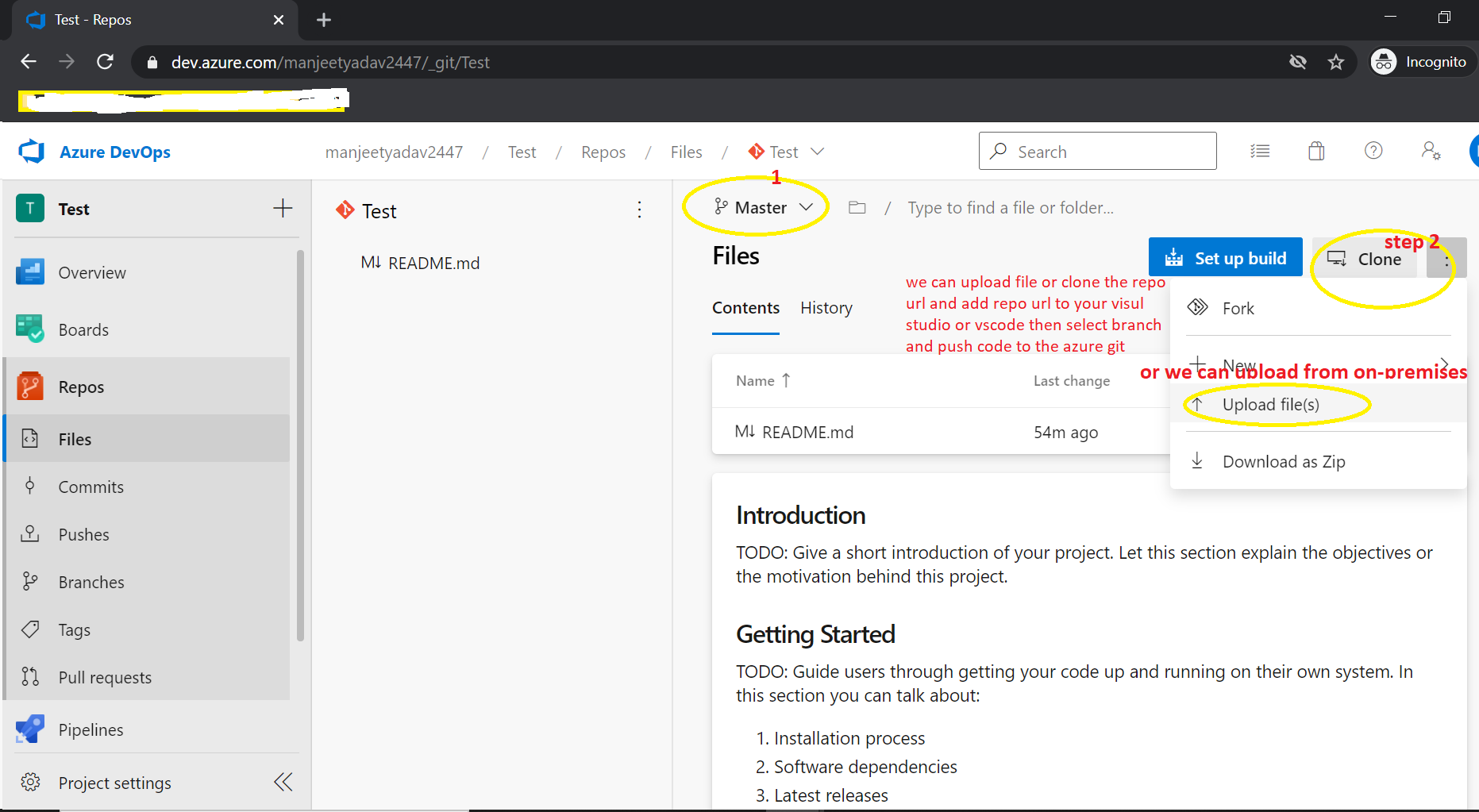
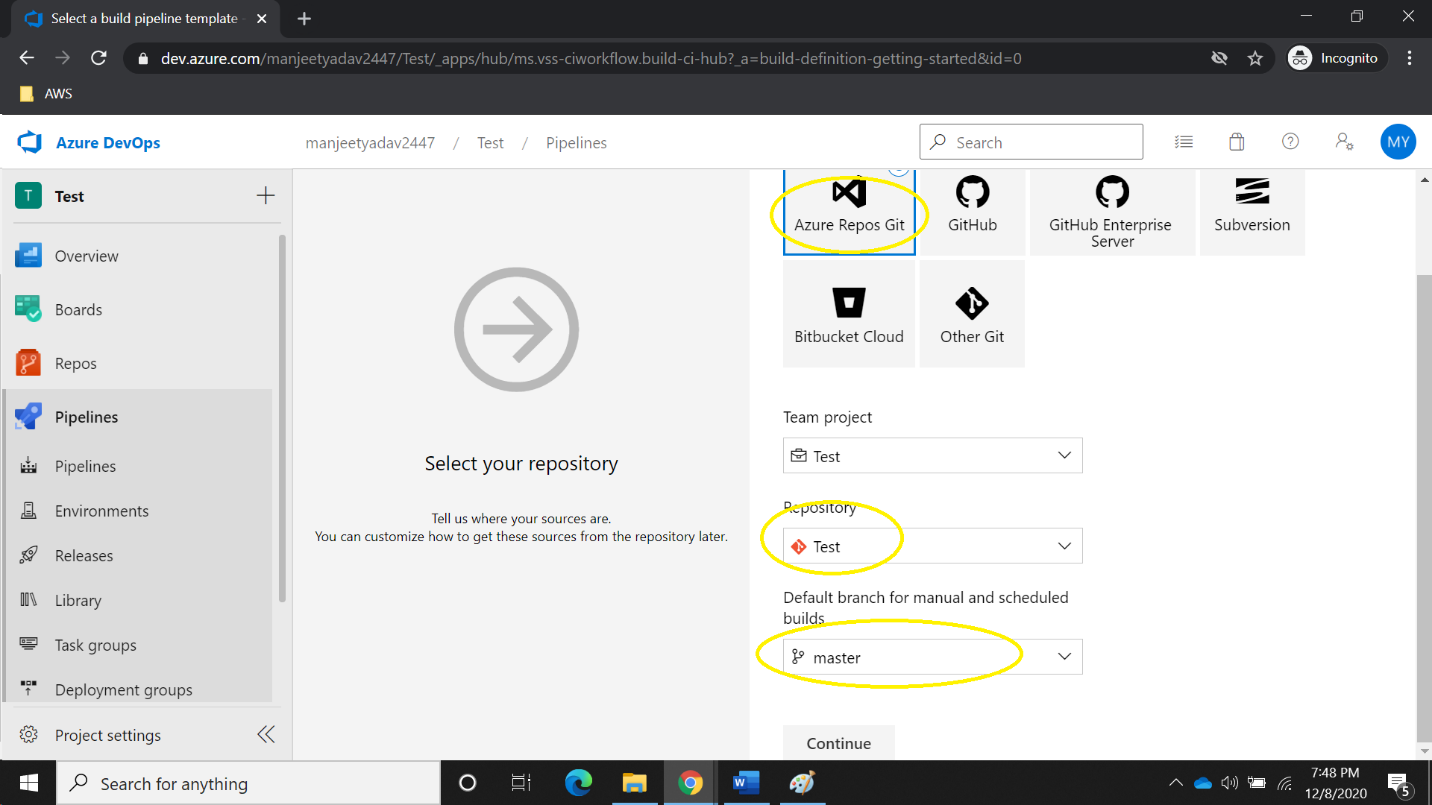
1. Steps how to create branches in azure git repo.



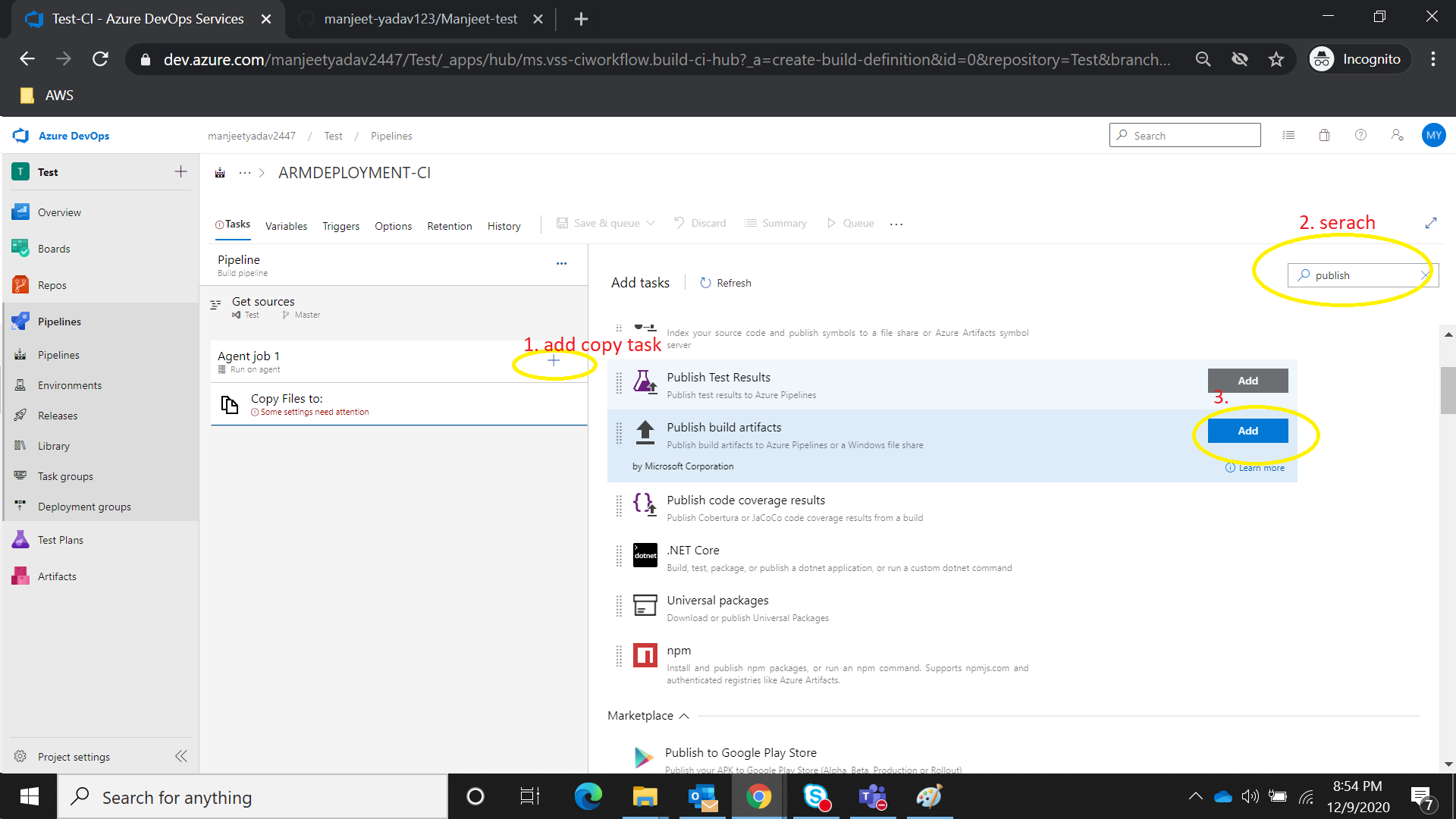
1. Please see the important note that I highlighted in red color(how to clone and push your code to azure git) –

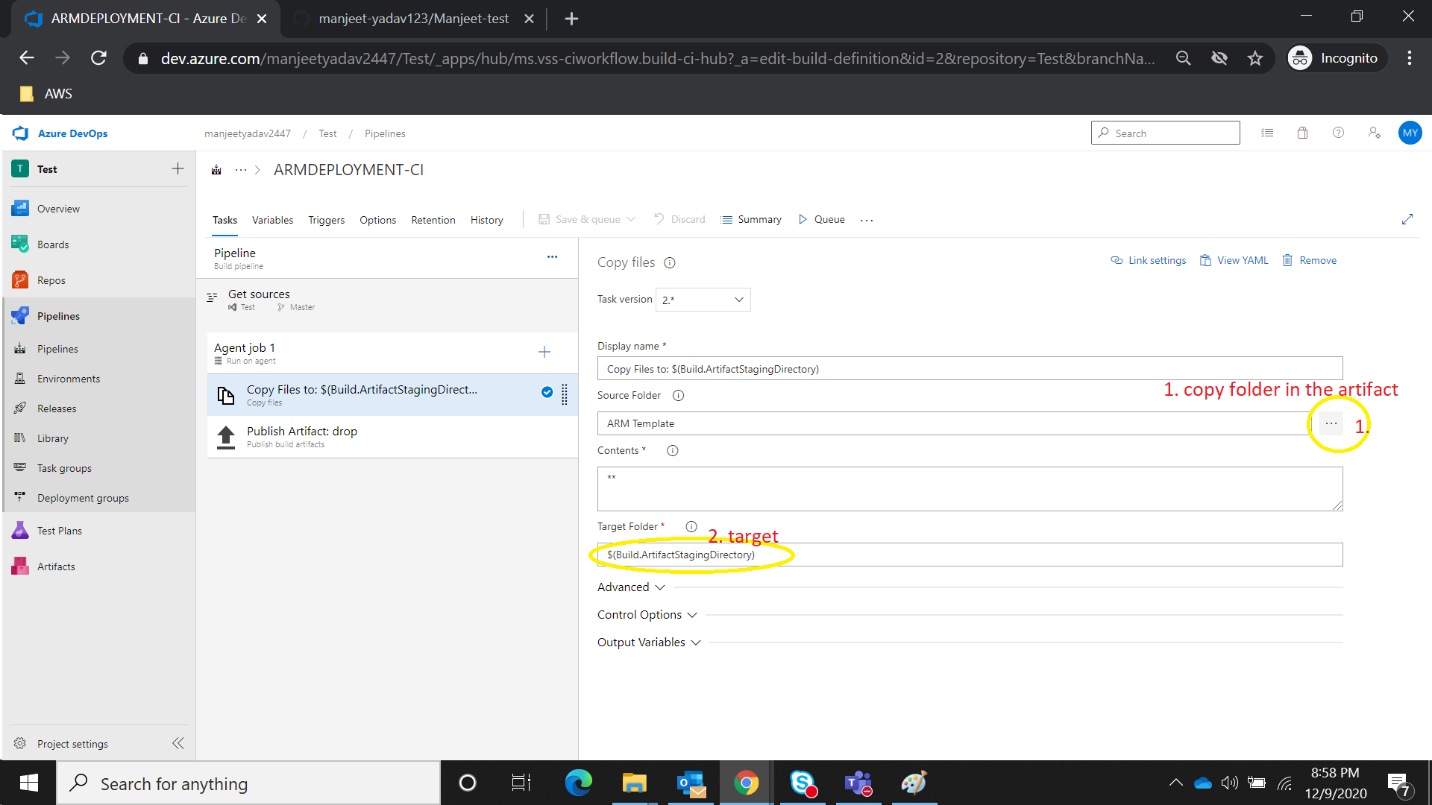


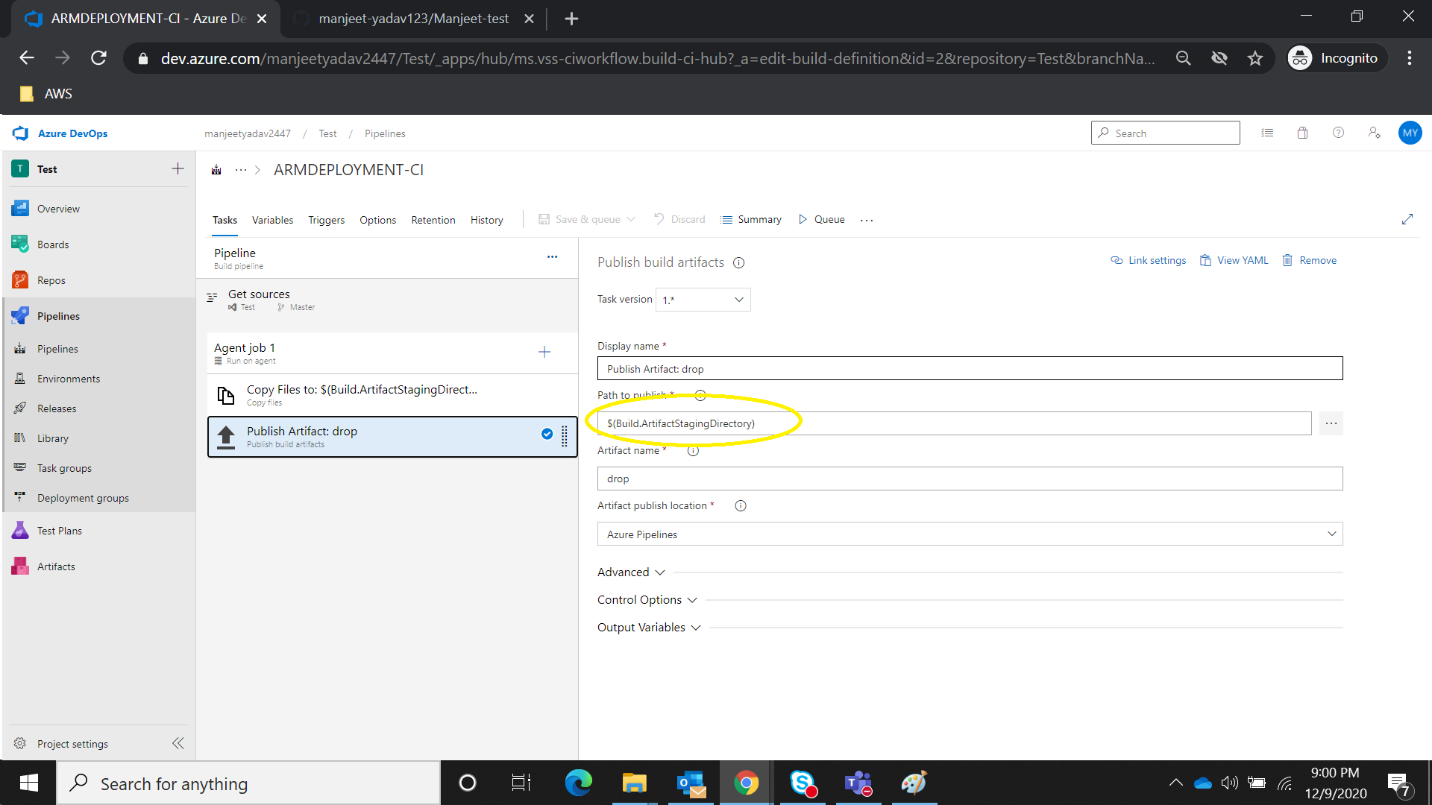
1. Steps for how to create build pipeline for dotnet Web App and Web Api project.

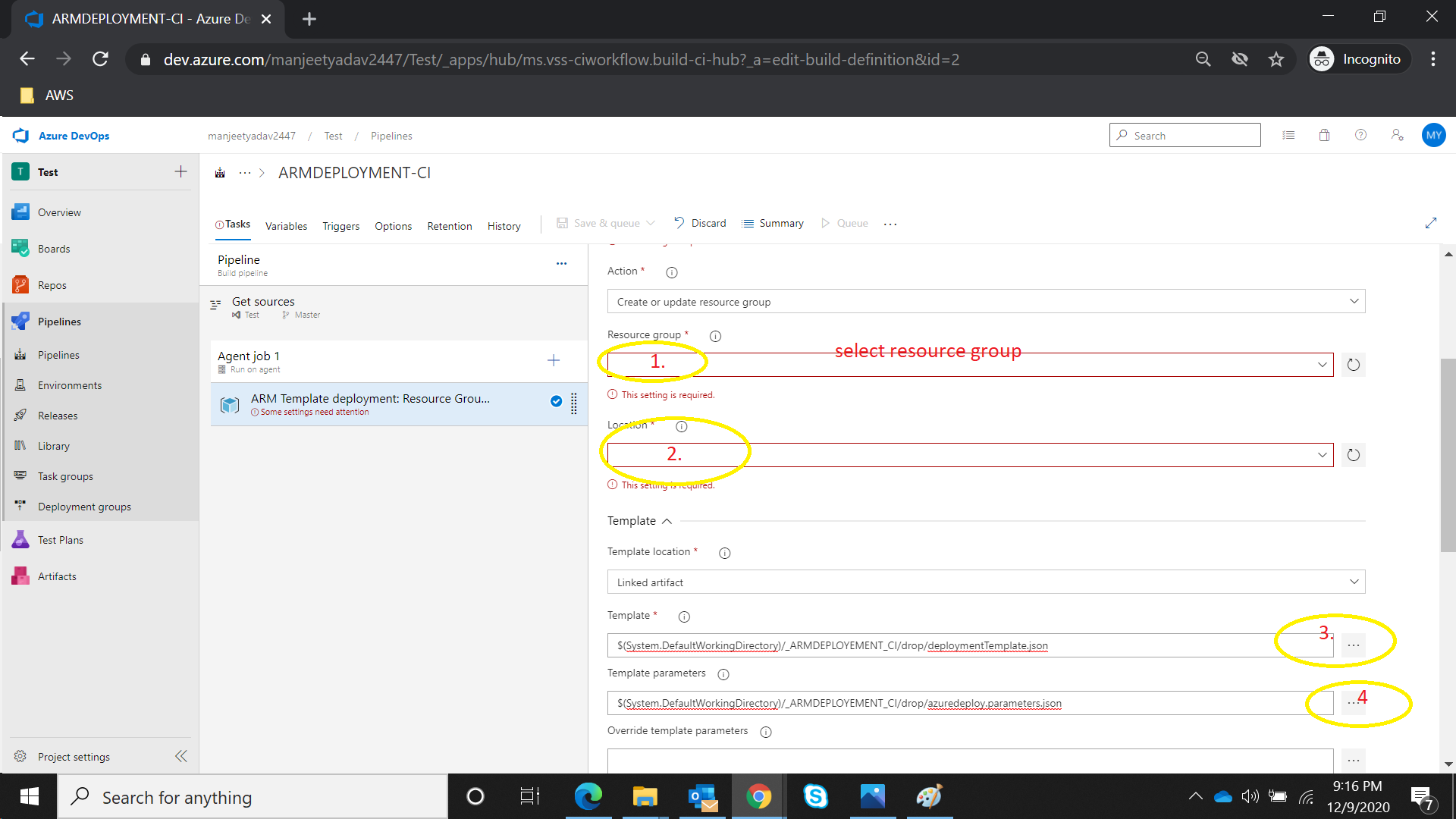


1. Add Task.





1. 
2. How to Automate IAAC using azure Devops pipeline.one more step missed in the Screenshot first of all I will have to select subscription.



1. Find ARM Deployment jsonfile.

{

"$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",

"contentVersion": "1.0.0.0",

"parameters": {

"myAdminUsername": {

"type": "string",

},

"MyAdministrator": {

"type": "securestring",

},

"dnsLabelPrefix": {

"type": "string",

"defaultValue": "[concat('vm-', uniqueString(resourceGroup().id))]",

},

"windowsOSVersion": {

"type": "string",

"defaultValue": "2019-Datacenter",

"allowedValues": [

"2012-Datacenter",

"2012-R2-Datacenter",

"2016-Datacenter",

"2019-Datacenter"

],

},

"location": {

"type": "string",

"defaultValue": "[resourceGroup().location]",

},

"vmSize": {

"type": "string",

"defaultValue": "Standard\_D2\_v3",

}

},

"variables": {

"storageAccountName": "[concat(uniquestring(resourceGroup().id), 'sawinvm')]",

"imagePublisher": "MicrosoftWindowsServer",

"imageOffer": "WindowsServer",

"nicName": "myVMNic",

"addressPrefix": "10.0.0.0/16",

"subnetName": "Subnet",

"subnetPrefix": "10.0.0.0/24",

"storageAccountType": "Standard\_LRS",

"publicIPAddressName": "myPublicIP",

"publicIPAddressType": "Dynamic",

"vmName": "SimpleWindowsVM",

"virtualNetworkName": "MyVNET",

"subnetRef": "[resourceId('Microsoft.Network/virtualNetworks/subnets', variables('virtualNetworkName'), variables('subnetName'))]",

"networkSecurityGroupName": "[concat(variables('subnetName'), '-nsg')]"

},

"resources": [

{

"type": "Microsoft.Storage/storageAccounts",

"name": "[variables('storageAccountName')]",

"apiVersion": "2019-06-01",

"location": "[parameters('location')]",

"kind": "Storage",

"sku": {

"name": "[variables('storageAccountType')]"

}

},

{

"apiVersion": "2019-11-01",

"type": "Microsoft.Network/publicIPAddresses",

"name": "[variables('publicIPAddressName')]",

"location": "[parameters('location')]",

"properties": {

"publicIPAllocationMethod": "[variables('publicIPAddressType')]",

"dnsSettings": {

"domainNameLabel": "[parameters('dnsLabelPrefix')]"

}

}

},

{

"comments": "Simple Network Security Group for subnet [variables('subnetName')]",

"type": "Microsoft.Network/networkSecurityGroups",

"apiVersion": "2019-11-01",

"name": "[variables('networkSecurityGroupName')]",

"location": "[parameters('location')]",

"properties": {

"securityRules": [

{

"name": "default-allow-3389",

"properties": {

"priority": 1000,

"access": "Allow",

"direction": "Inbound",

"destinationPortRange": "3389",

"protocol": "Tcp",

"sourceAddressPrefix": "\*",

"sourcePortRange": "\*",

"destinationAddressPrefix": "\*"

}

}

]

}

},

{

"apiVersion": "2019-11-01",

"type": "Microsoft.Network/virtualNetworks",

"name": "[variables('virtualNetworkName')]",

"location": "[parameters('location')]",

"dependsOn": [

"[variables('networkSecurityGroupName')]"

],

"properties": {

"addressSpace": {

"addressPrefixes": [

"[variables('addressPrefix')]"

]

},

"subnets": [

{

"name": "[variables('subnetName')]",

"properties": {

"addressPrefix": "[variables('subnetPrefix')]",

"networkSecurityGroup": {

"id": "[resourceId('Microsoft.Network/networkSecurityGroups', variables('networkSecurityGroupName'))]"

}

}

}

]

}

},

{

"apiVersion": "2019-11-01",

"type": "Microsoft.Network/networkInterfaces",

"name": "[variables('nicName')]",

"location": "[parameters('location')]",

"dependsOn": [

"[variables('publicIPAddressName')]",

"[variables('virtualNetworkName')]"

],

"properties": {

"ipConfigurations": [

{

"name": "ipconfig1",

"properties": {

"privateIPAllocationMethod": "Dynamic",

"publicIPAddress": {

"id": "[resourceId('Microsoft.Network/publicIPAddresses', variables('publicIPAddressName'))]"

},

"subnet": {

"id": "[variables('subnetRef')]"

}

}

}

]

}

},

{

"apiVersion": "2019-12-01",

"type": "Microsoft.Compute/virtualMachines",

"name": "[variables('vmName')]",

"location": "[parameters('location')]",

"dependsOn": [

"[variables('storageAccountName')]",

"[variables('nicName')]"

],

"properties": {

"hardwareProfile": {

"vmSize": "[parameters('vmSize')]"

},

"osProfile": {

"computerName": "[variables('vmName')]",

"adminUsername": "[parameters('MyAdministrator')]",

"adminPassword": "[parameters('MyAdministrator')]"

},

"storageProfile": {

"imageReference": {

"publisher": "[variables('imagePublisher')]",

"offer": "[variables('imageOffer')]",

"sku": "[parameters('windowsOSVersion')]",

"version": "latest"

},

"osDisk": {

"createOption": "FromImage"

},

"dataDisks": [

{

"diskSizeGB": 1023,

"lun": 0,

"createOption": "Empty"

}

]

},

"networkProfile": {

"networkInterfaces": [

{

"id": "[resourceId('Microsoft.Network/networkInterfaces',variables('nicName'))]"

}

]

},

"diagnosticsProfile": {

"bootDiagnostics": {

"enabled": true,

"storageUri": "[reference(variables('storageAccountName')).primaryEndpoints.blob]"

}

}

}

}

]

}

1. Please find Parameter file .

{

"$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentParameters.json#",

"contentVersion": "1.0.0.0",

"parameters": {

"myAdminUsername": {

"value": "MyAdministrator"

},

"myAdminPassword": {

"reference": {

"keyVault": {

"id": "/subscriptions/edf192ff-xxxx-xxxx-xxxx-xxxxxxxxxxxx/resourceGroups/AzureKeyVaultDemo/providers/Microsoft.KeyVault/vaults/MyUniqueKeyVaultName"

},

"secretName": "MyAdminPassword"

}

}

}

}