**YAML**

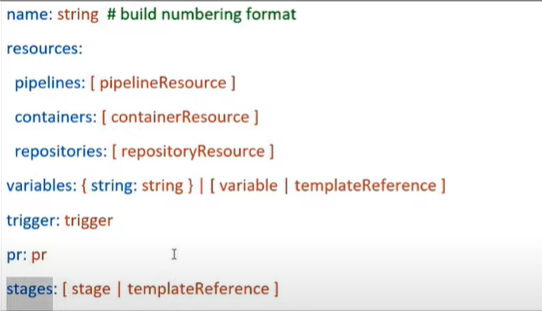
YAML: **Y**et **A**nother **M**arkup **L**anguage

Build Pipeline = CI

Release Pipeline = CD

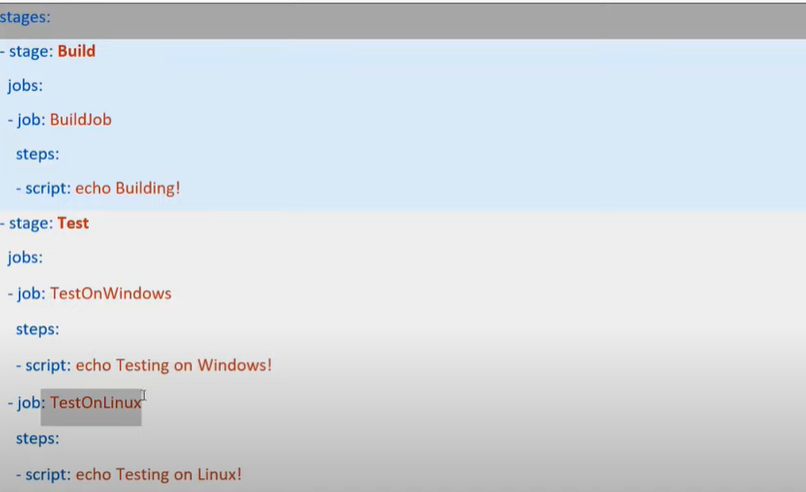
YAML pipeline = CI + CD

**YAML Format:**

****

Minimum 1 stage will be there

Stage: it is logical boundary in pipeline. Each stage contains one or more jobs



Here we see that one stage Test has 2 jobs

Job: Job is collection of linear series of step

YAML Pipeline

Stage

Job

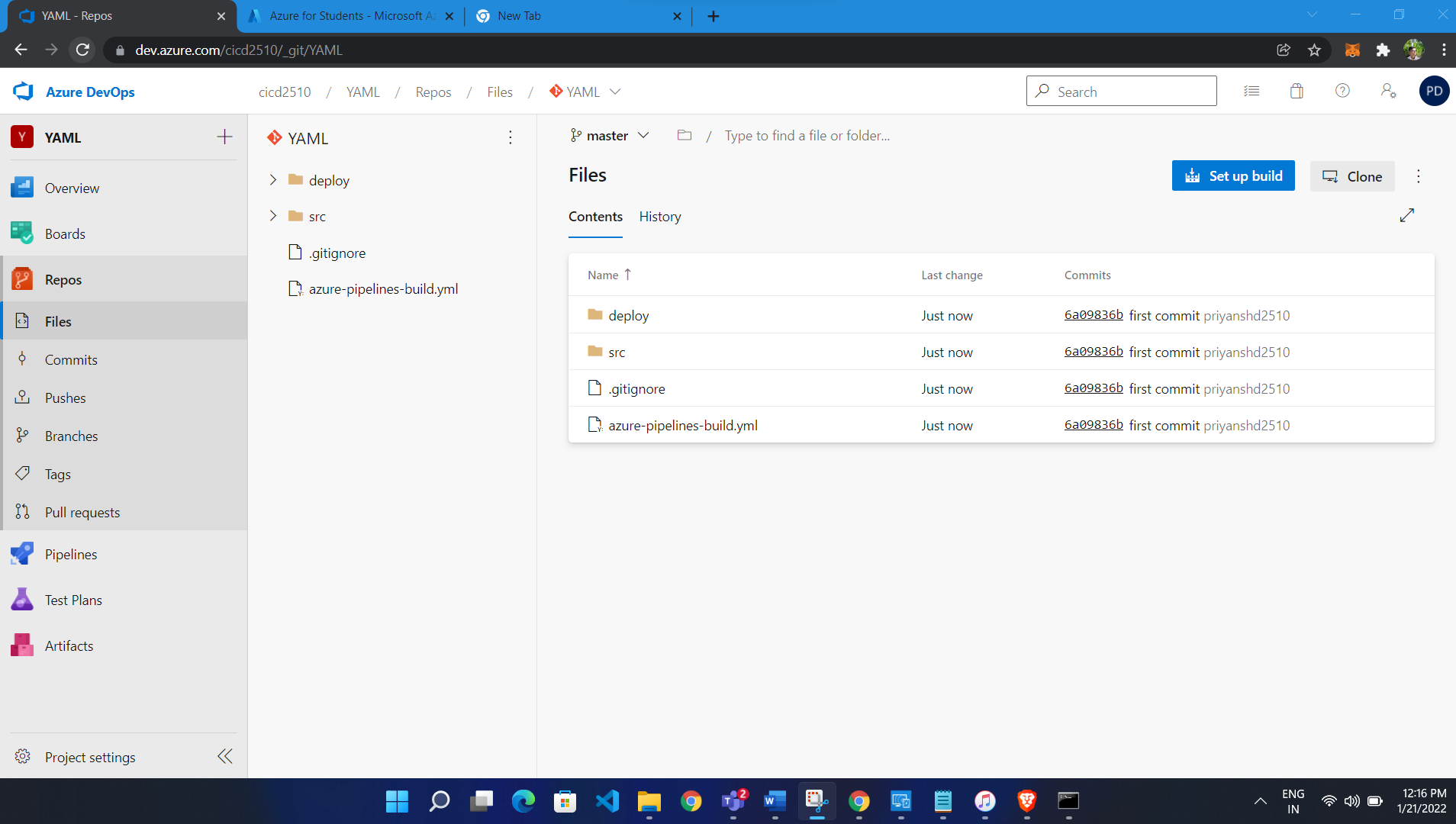
Script/task

Every job has Agent

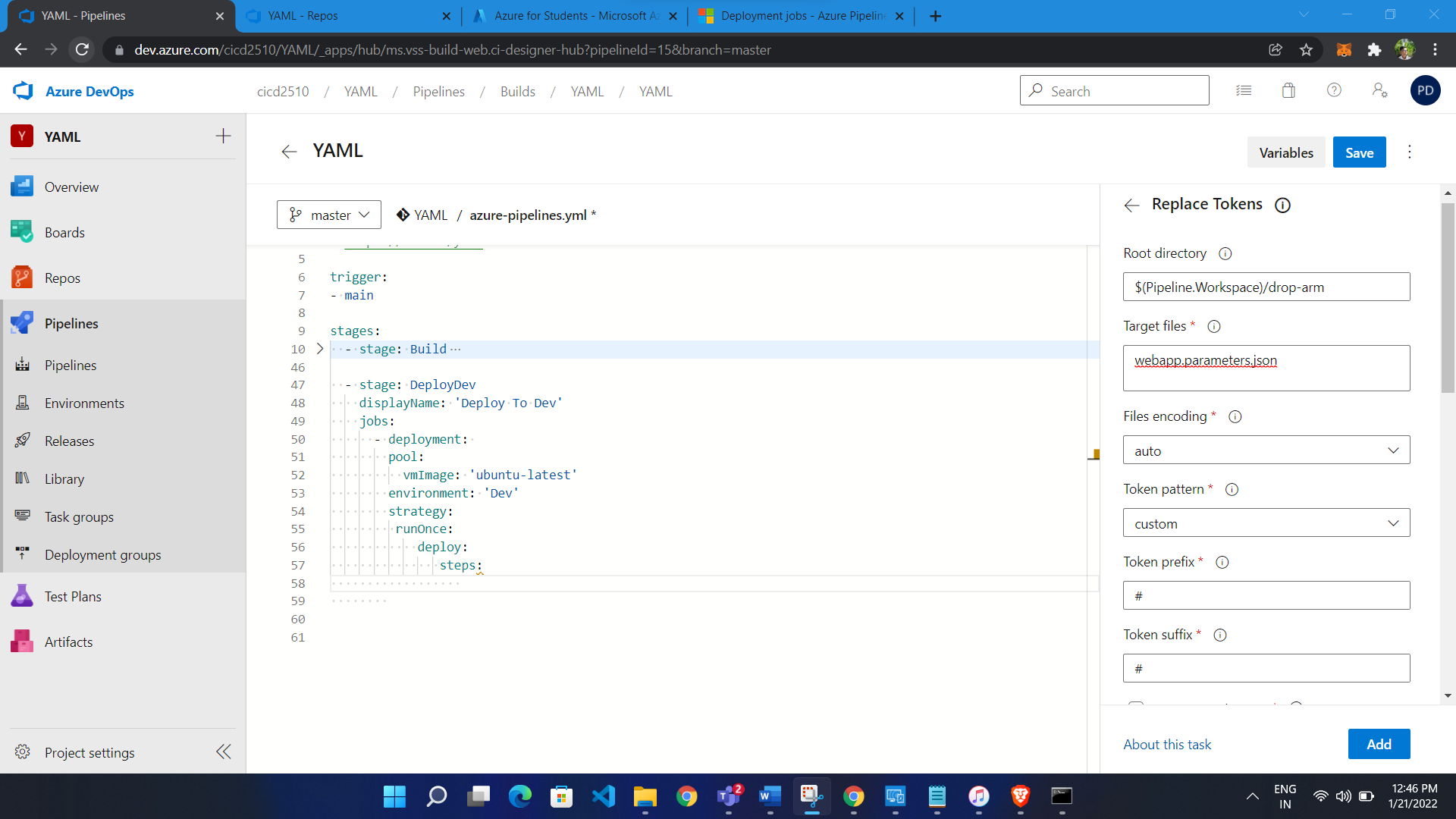
YAML:

|  |
| --- |
| # Starter pipeline  # Start with a minimal pipeline that you can customize to build and deploy your code.  # Add steps that build, run tests, deploy, and more:  # https://aka.ms/yaml  trigger:  - main  stages:    - stage: Build      jobs:        - job: BuildWebApp          variables:            buildConfiguration: 'Release'          pool:            vmImage: 'ubuntu-latest'          steps:          - task: UseDotNet@2            inputs:              packageType: 'sdk'              version: '5.x'          - task: DotNetCoreCLI@2            displayName: 'Build Application'            inputs:              command: 'publish'              publishWebProjects: true              arguments: '--configuration $(buildConfiguration) --output $(Build.BinariesDirectory)/$(Build.BuildId)'              modifyOutputPath: false              workingDirectory: 'src'            - task: PublishBuildArtifacts@1            displayName: 'Publish Web Artifact'            inputs:              PathtoPublish: '$(Build.BinariesDirectory)/$(Build.BuildId)'              ArtifactName: 'drop-$(Build.BuildId)'              publishLocation: 'Container'          - task: PublishBuildArtifacts@1            displayName: 'Publish ARM Artifact'            inputs:              PathtoPublish: 'deploy/'              ArtifactName: 'drop-arm'              publishLocation: 'Container'    - stage: DeployDev      displayName: 'Deploy To Dev'      variables:           - group: 'YAML-Releases-Dev'      jobs:        - deployment:          pool:            vmImage: 'ubuntu-latest'          environment: 'Dev'          strategy:           runOnce:              deploy:                 steps:                  - task: replacetokens@4                    inputs:                      rootDirectory: '$(Pipeline.Workspace)/drop-arm'                      targetFiles: 'webapp.parameters.json'                      encoding: 'auto'                      tokenPattern: 'custom'                      tokenPrefix: '#'                      tokenSuffix: '#'                      writeBOM: false                      actionOnMissing: 'warn'                      keepToken: false                      actionOnNoFiles: 'continue'                      enableTransforms: false                      useLegacyPattern: false                      enableTelemetry: true                    - task: AzureResourceManagerTemplateDeployment@3                    inputs:                      deploymentScope: 'Resource Group'                      azureResourceManagerConnection: 'connection'                      subscriptionId: 'af173fdc-9c6c-4fa7-bab2-69d025b4142a'                      action: 'Create Or Update Resource Group'                      resourceGroupName: 'Test-DEV'                      location: 'East US'                      templateLocation: 'Linked artifact'                      csmFile: '$(Pipeline.Workspace)/drop-arm/webapp.json'                      csmParametersFile: '$(Pipeline.Workspace)/drop-arm/webapp.parameters.json'                      deploymentMode: 'Incremental'                  - task: AzureRmWebAppDeployment@4                    inputs:                      ConnectionType: 'AzureRM'                      azureSubscription: 'connection'                      appType: 'webApp'                      WebAppName: '$(arm\_webApp)'                      packageForLinux: '$(Pipeline.workspace)/\*\*/\*.zip'    - stage: DeployTest      displayName: 'Deploy To Test'      variables:           - group: 'YAML-Releases-Test'      jobs:        - deployment:          pool:            vmImage: 'ubuntu-latest'          environment: 'Test'          strategy:           runOnce:              deploy:                 steps:                  - task: replacetokens@4                    inputs:                      rootDirectory: '$(Pipeline.Workspace)/drop-arm'                      targetFiles: 'webapp.parameters.json'                      encoding: 'auto'                      tokenPattern: 'custom'                      tokenPrefix: '#'                      tokenSuffix: '#'                      writeBOM: false                      actionOnMissing: 'warn'                      keepToken: false                      actionOnNoFiles: 'continue'                      enableTransforms: false                      useLegacyPattern: false                      enableTelemetry: true                    - task: AzureResourceManagerTemplateDeployment@3                    inputs:                      deploymentScope: 'Resource Group'                      azureResourceManagerConnection: 'connection'                      subscriptionId: 'af173fdc-9c6c-4fa7-bab2-69d025b4142a'                      action: 'Create Or Update Resource Group'                      resourceGroupName: 'Test-Test'                      location: 'East US'                      templateLocation: 'Linked artifact'                      csmFile: '$(Pipeline.Workspace)/drop-arm/webapp.json'                      csmParametersFile: '$(Pipeline.Workspace)/drop-arm/webapp.parameters.json'                      deploymentMode: 'Incremental'                  - task: AzureRmWebAppDeployment@4                    inputs:                      ConnectionType: 'AzureRM'                      azureSubscription: 'connection'                      appType: 'webApp'                      WebAppName: '$(arm\_webApp)'                      packageForLinux: '$(Pipeline.workspace)/\*\*/\*.zip' |

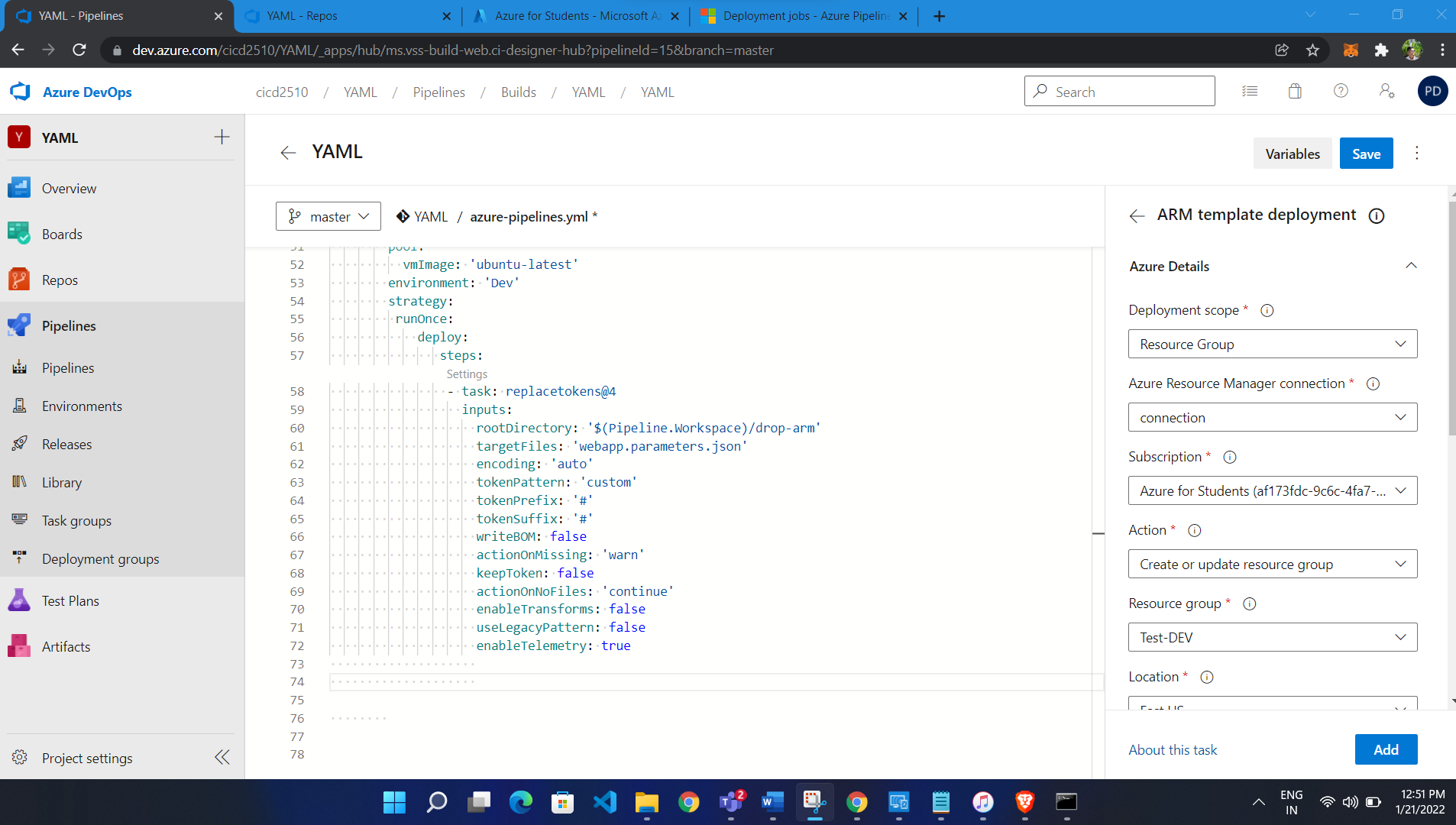
Configuration:



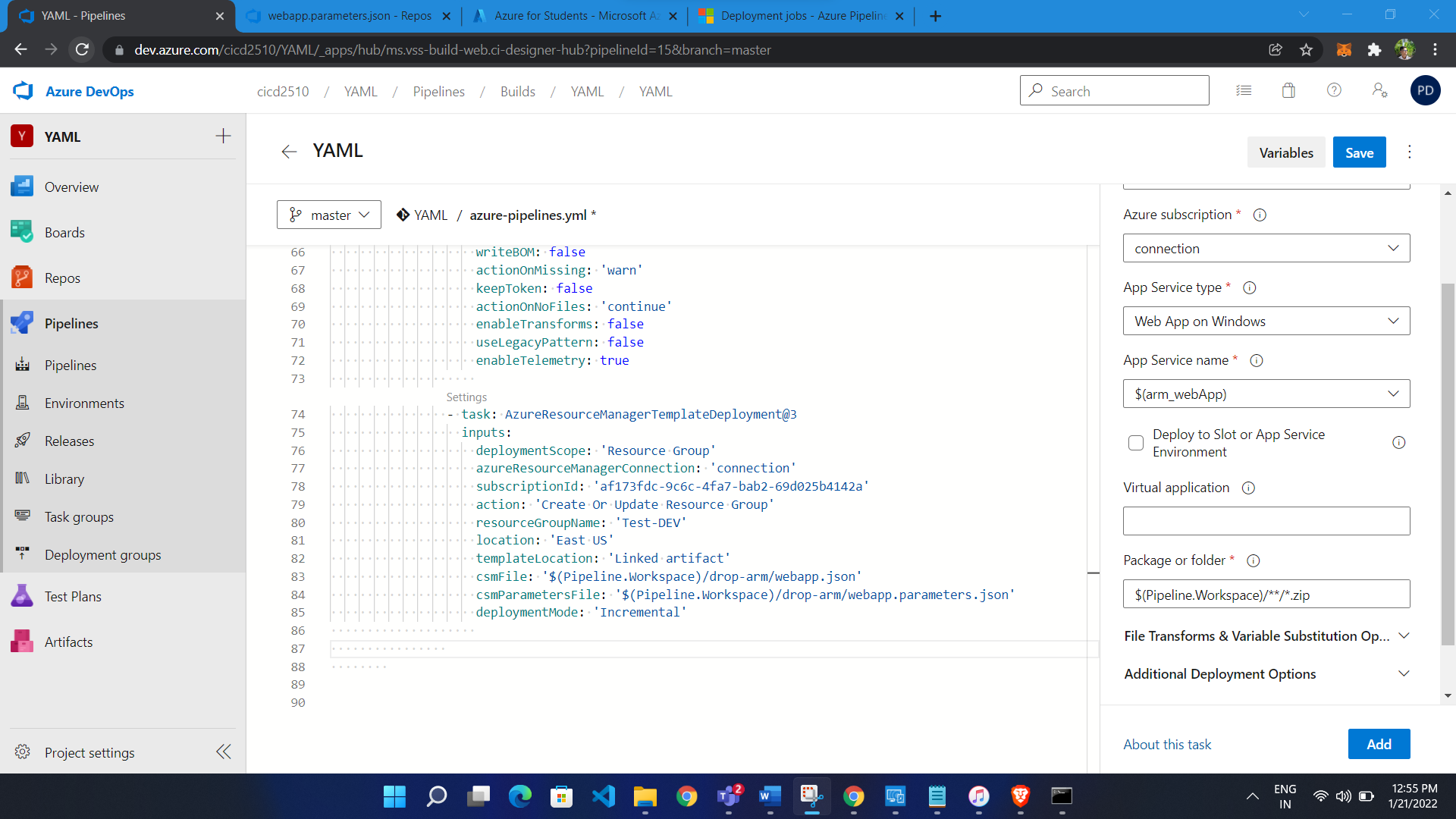
Add replace token



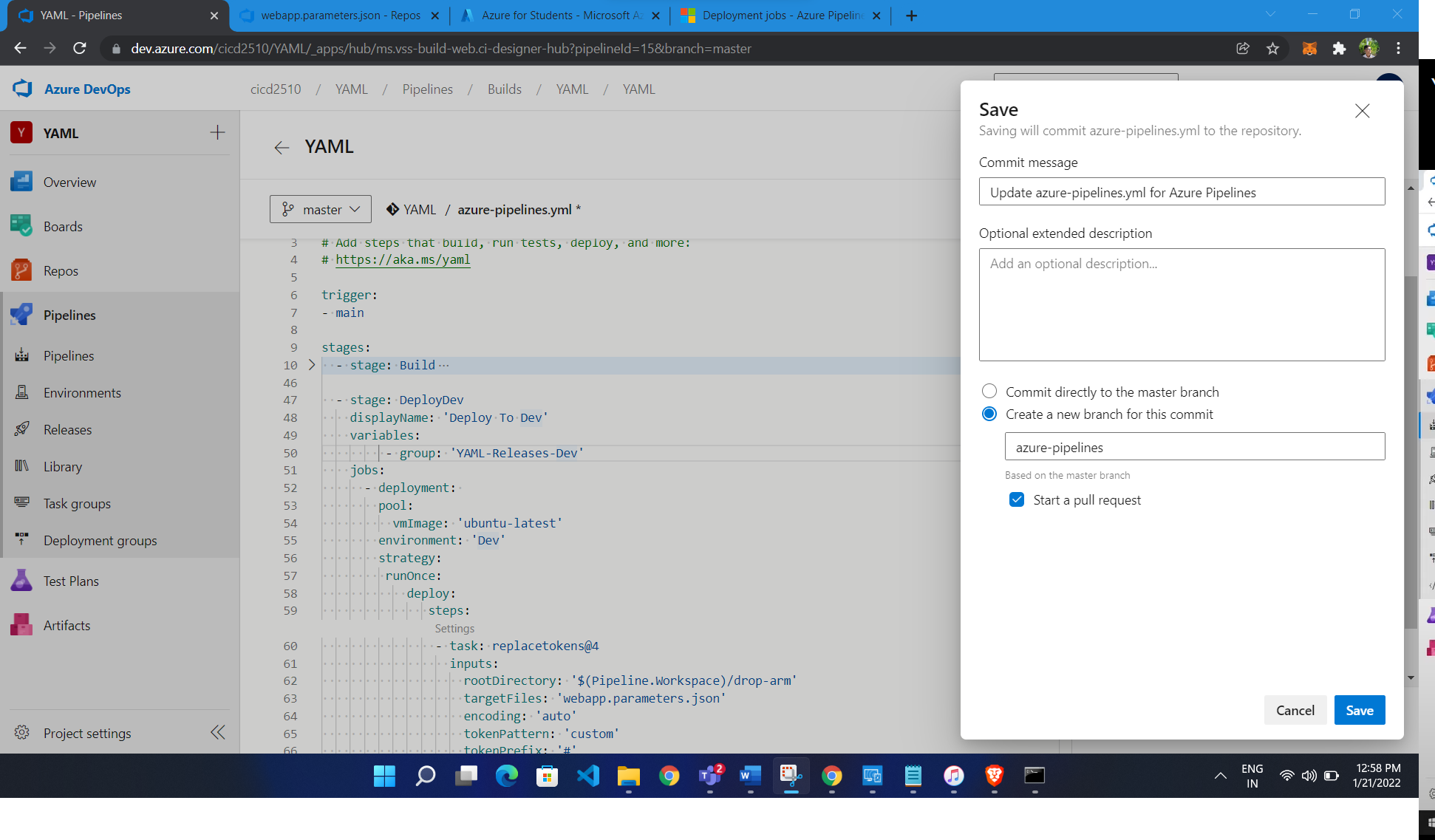
Add ARM Template Deployment



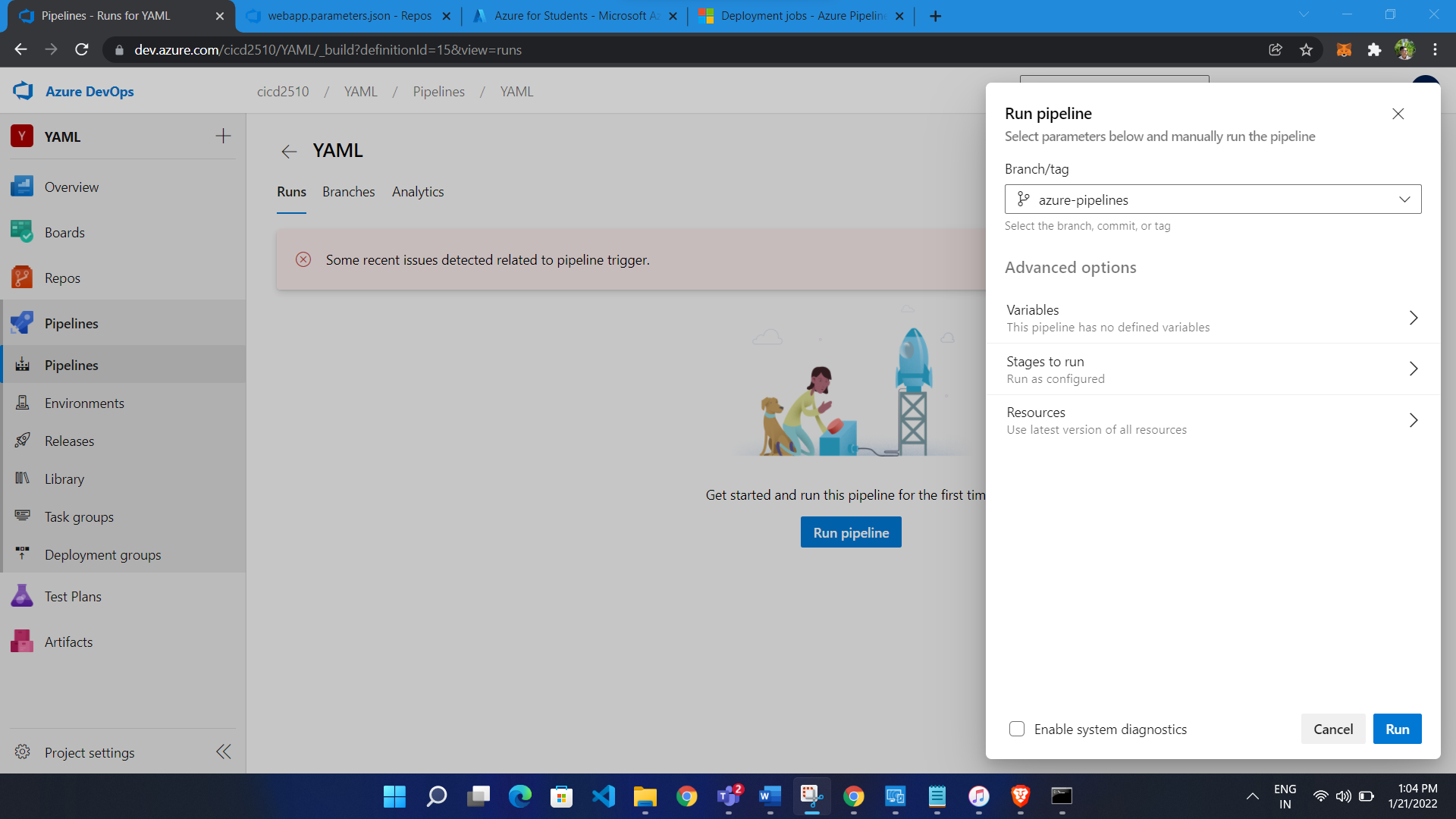
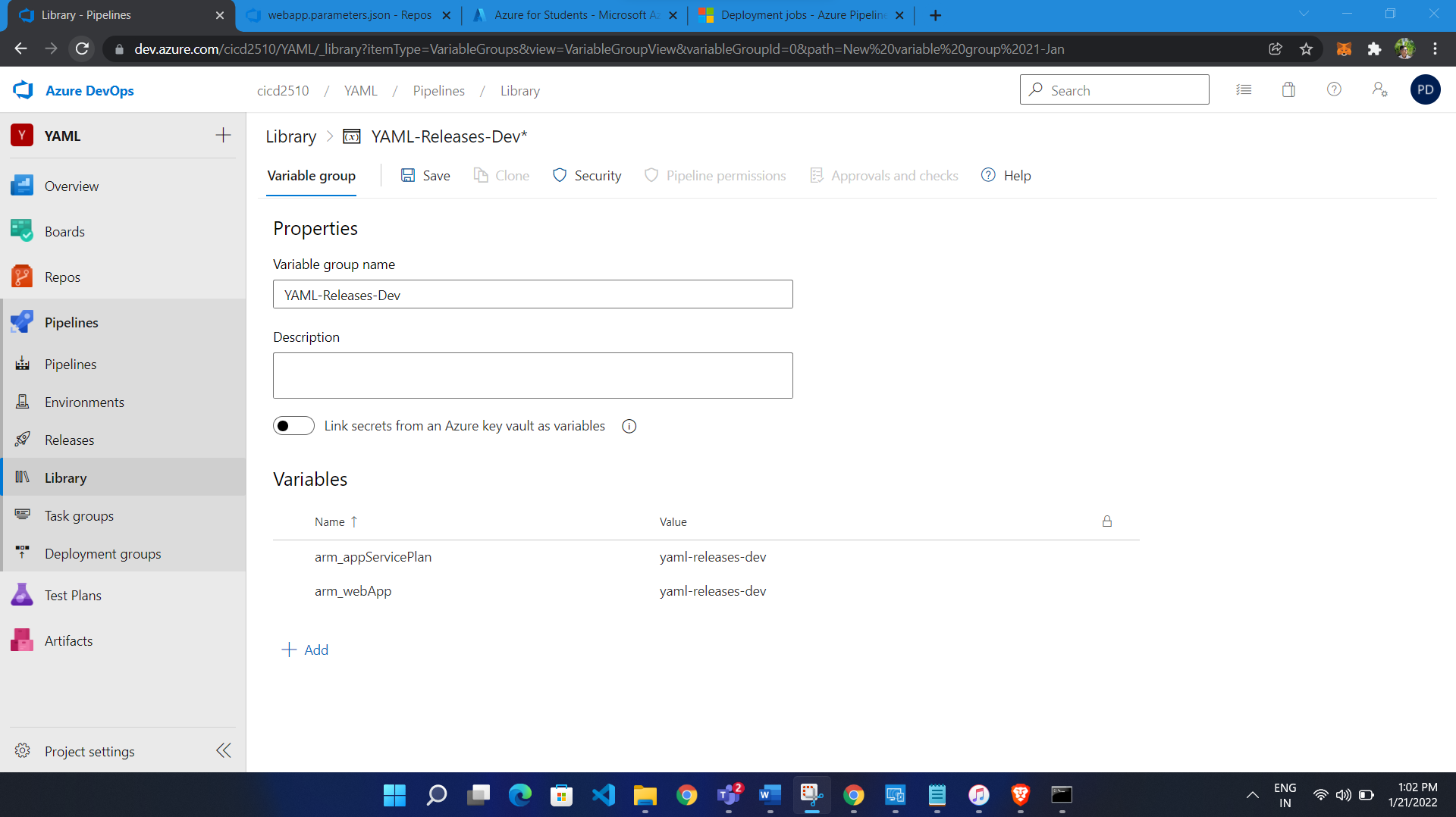
Add deploy to app service

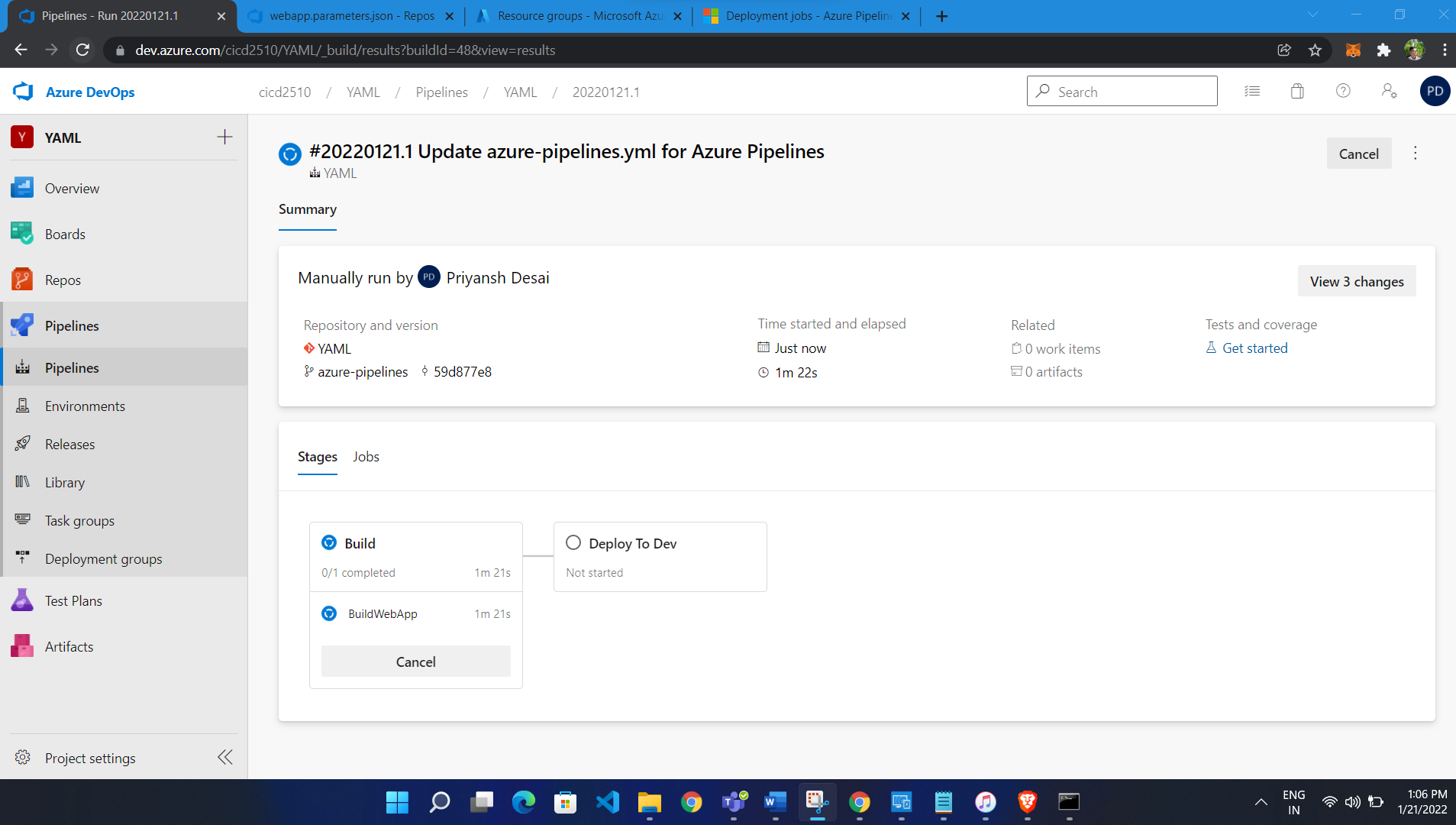


Save to new ranch

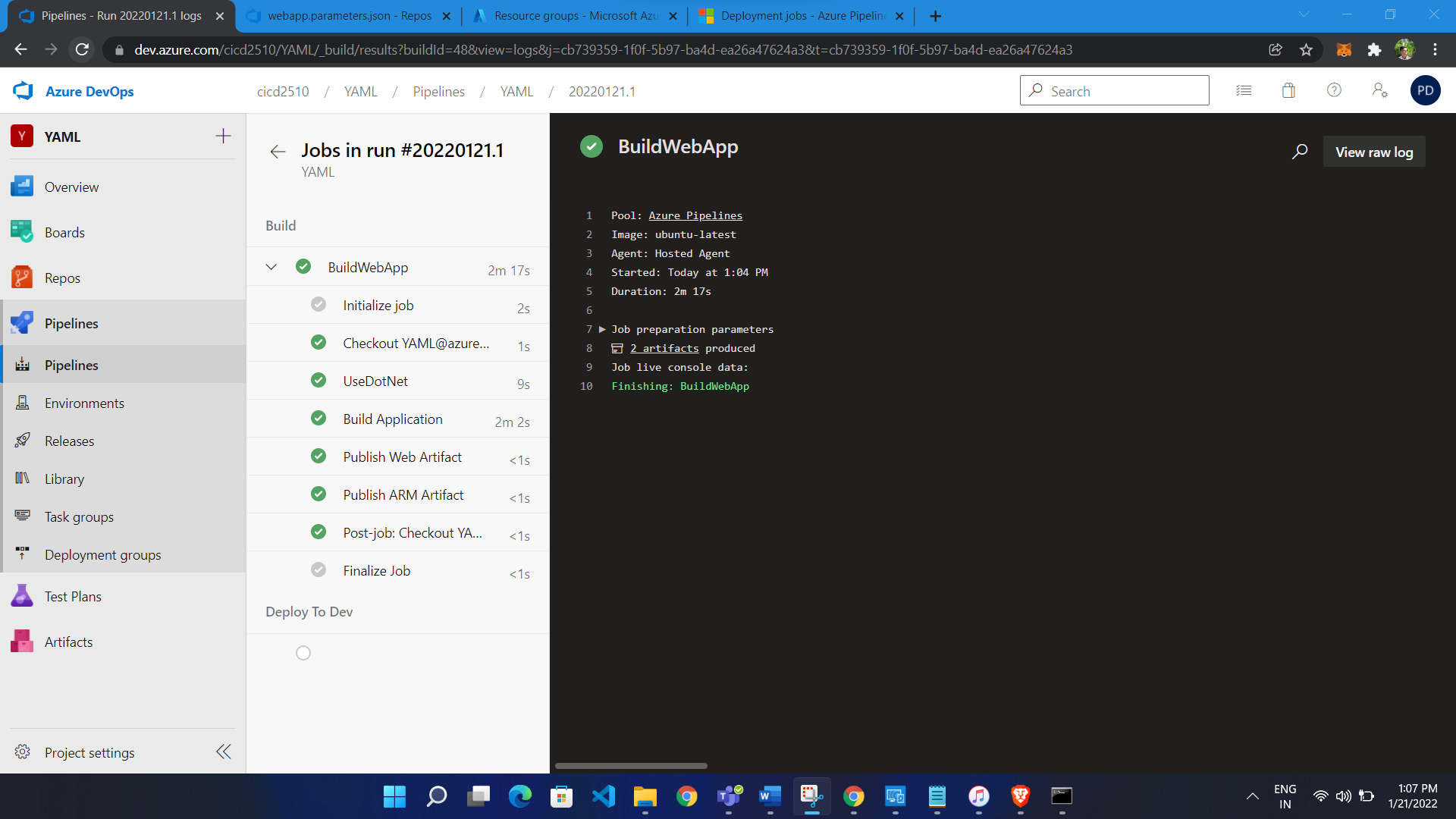


Add Variable Group

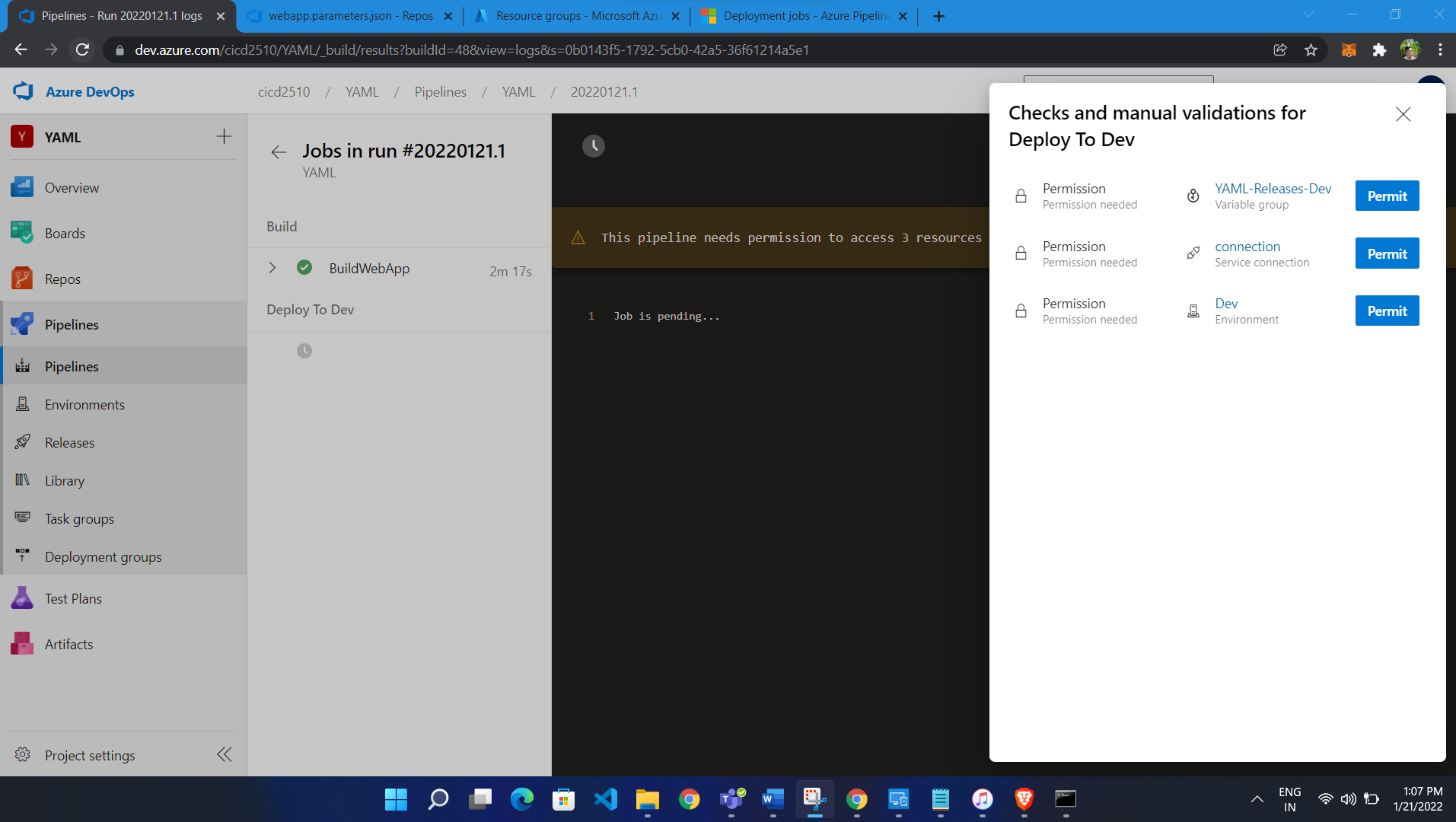




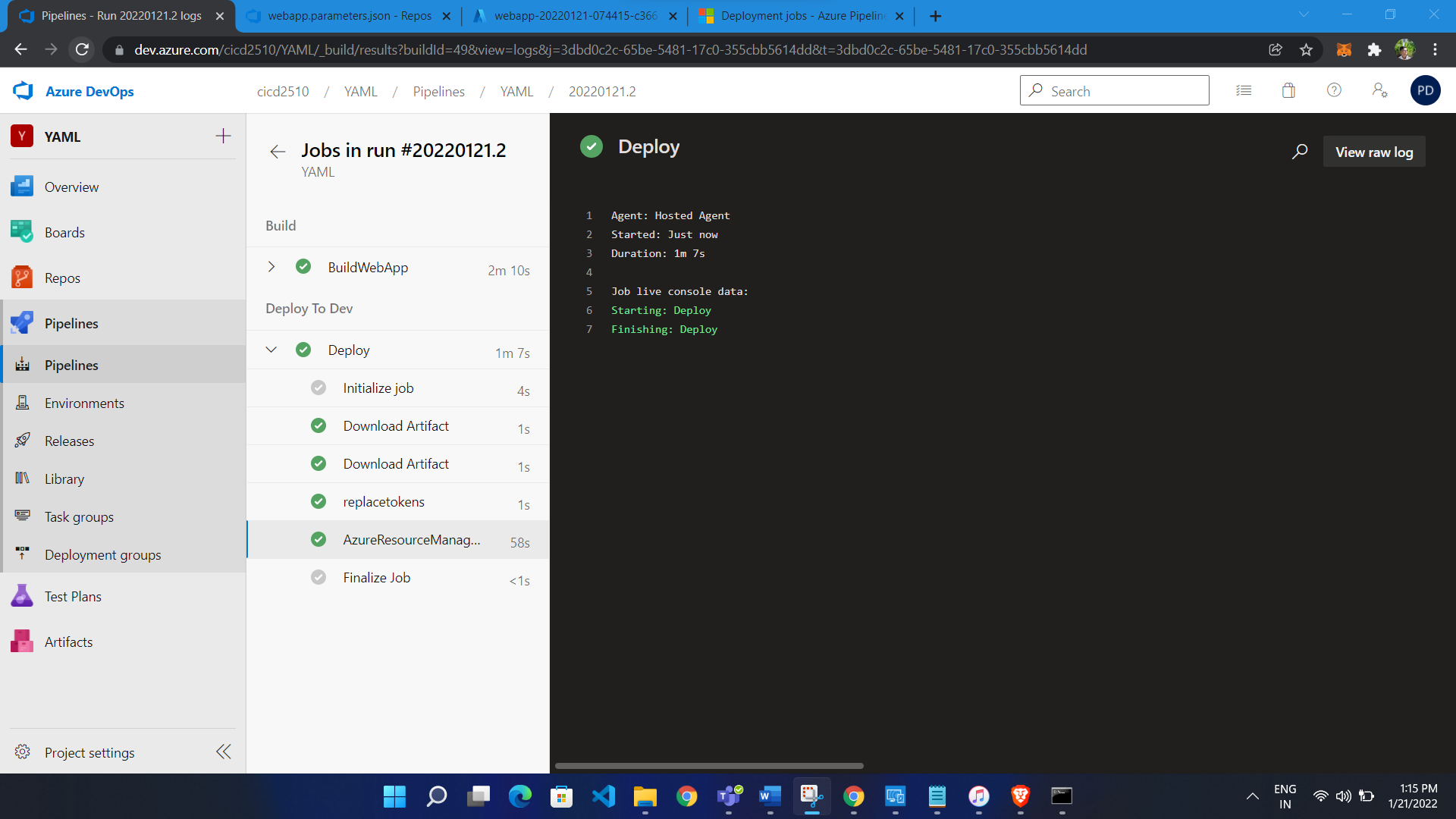
Build successfully



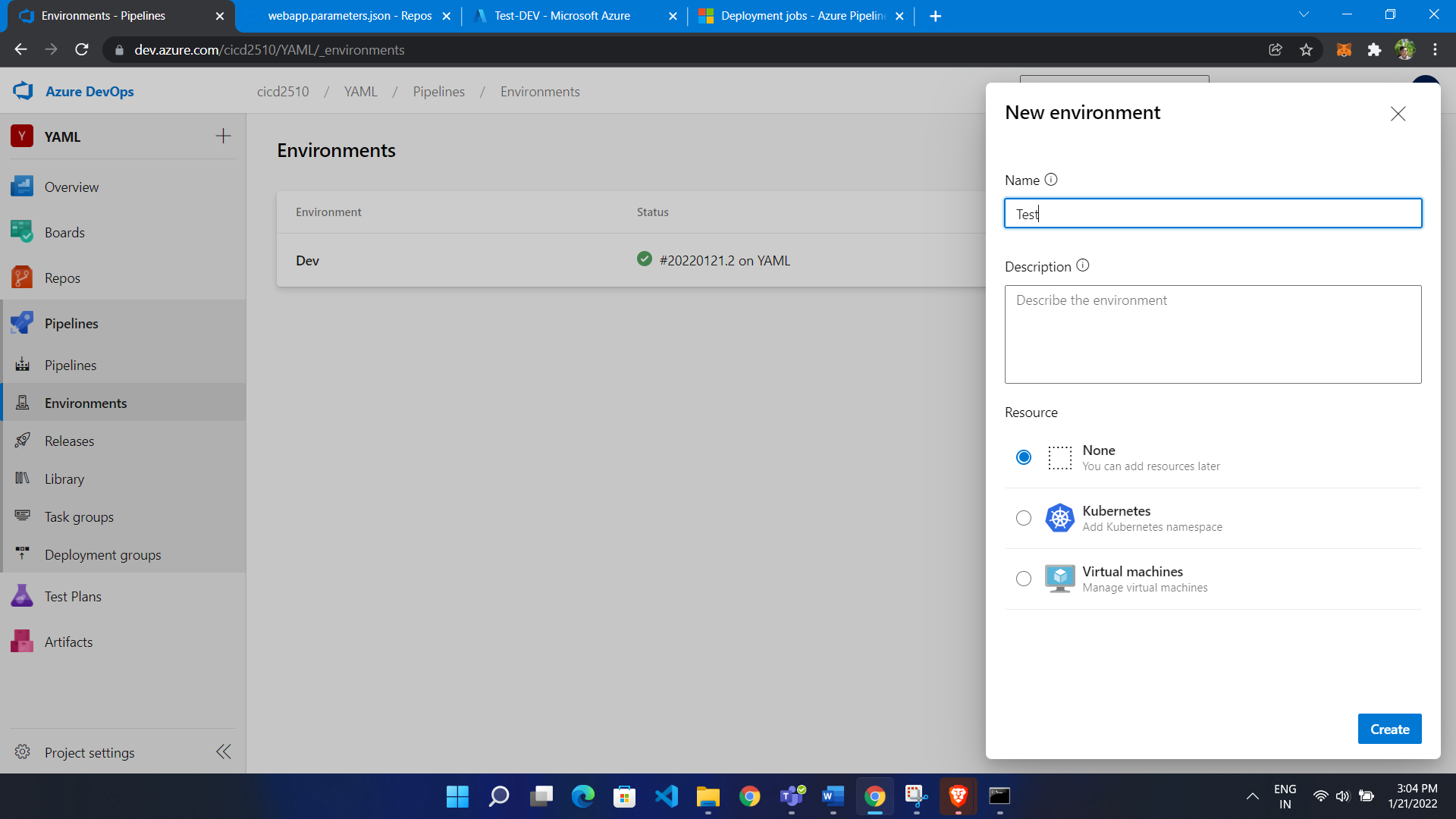
Give permission



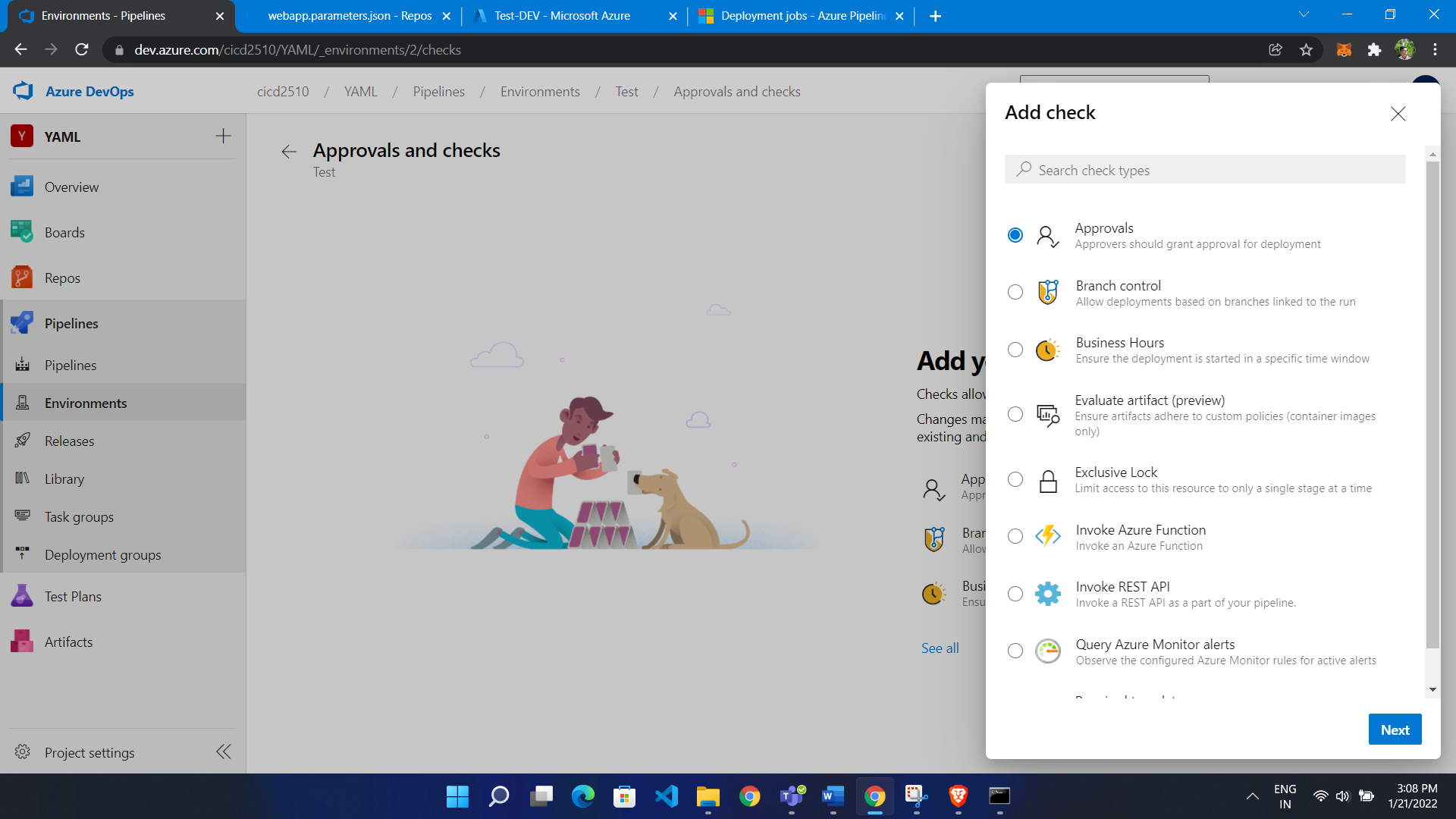
Release successfully

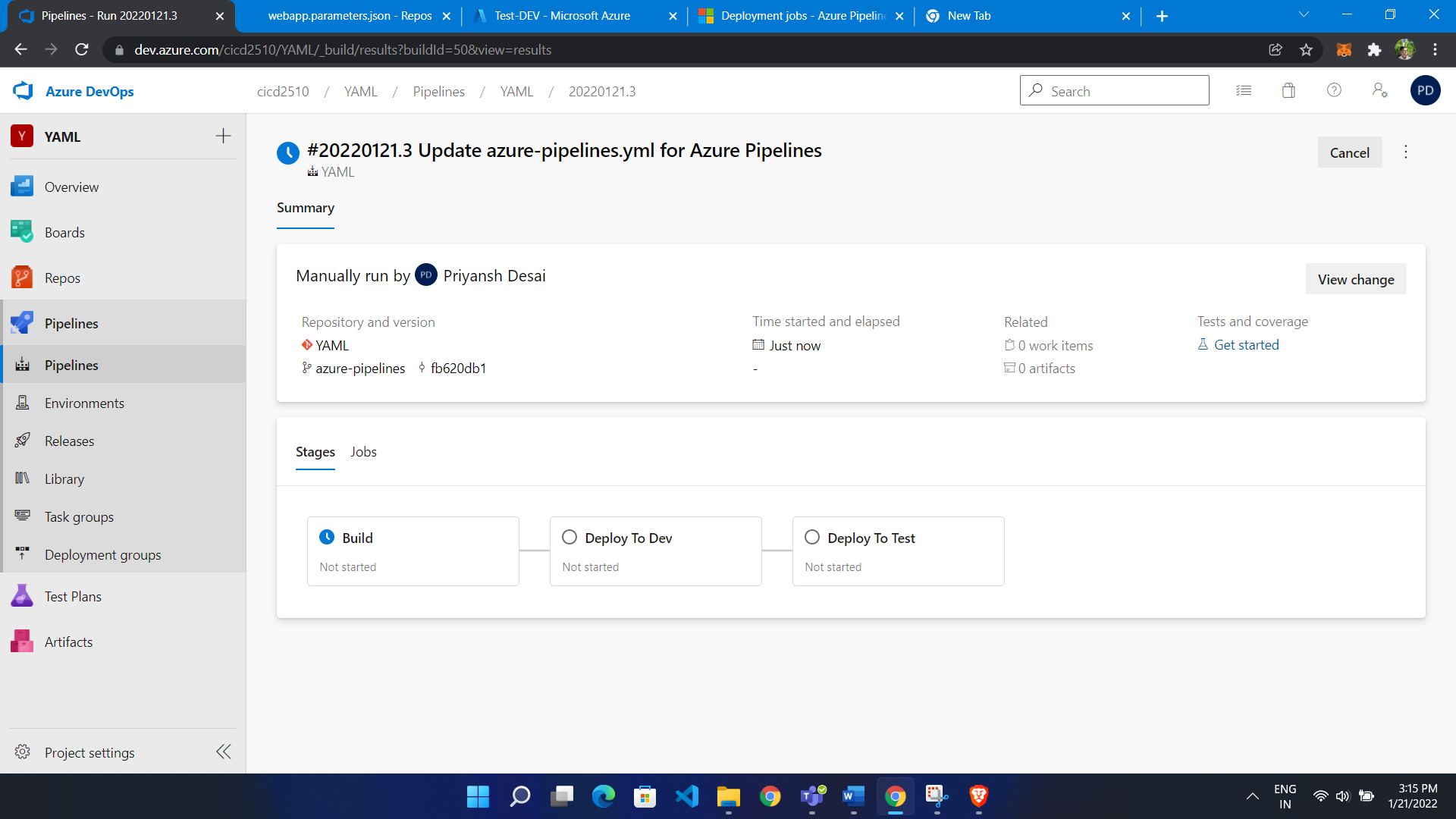


Create new Environment Test:



Set Approval





Here gave approvals:

