**SAMPLE PIPELINE:**

resources:

repositories:

- repository: myrepo1

type: git

name: projectname/reponame

- repository: myrepo2

type: git

name: projectname2/reponame2

trigger:

- master

pool:

vmImage: 'windows-latest'

variables:

solution: '\*\*/\*.sln'

jobs:

- job: Build

steps:

- checkout: myrepo1

- checkout: myrepo2

- task: nugetinstaller

- task: nugetRestore

- task: Prepare analysis on sonarqube

- task: Build the application using msbuild

- task: Run Code Analysis

- task: Publish Quality Gate result

- task: Publish the build arifact to JFrog lib

**MULTI AGENT PIPELINE:**

variables:

${{ eq(variables['Build.SourceBranch'], 'release/\*') }}:

release\_build: True

stages:

- stage: normal

condition: eq(variables['release\_build'], False)

pool:

vmImage: 'windows-latest'

jobs:

- job: Builds

steps:

- template: build.yaml

- stage: release

condition: eq(variables['release\_build'], True)

pool: My-agent

jobs:

- job: Builds

steps:

- template: build.yaml

**.NET core ARM TEMPLATE CI:**

# ASP.NET Core (.NET Framework)

# Build and test ASP.NET Core projects targeting the full .NET Framework.

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

# https://docs.microsoft.com/azure/devops/pipelines/languages/dotnet-core

trigger:

- main

pool:

  vmImage: 'windows-latest'

variables:

  solution: '\*\*/\*.sln'

  buildPlatform: 'Any CPU'

  buildConfiguration: 'Release'

steps:

- task: NuGetToolInstaller@1

- task: NuGetCommand@2

  inputs:

    restoreSolution: '$(solution)'

- task: VSBuild@1

  inputs:

    solution: '$(solution)'

    msbuildArgs: '/p:DeployOnBuild=true /p:WebPublishMethod=Package /p:PackageAsSingleFile=true /p:SkipInvalidConfigurations=true /p:DesktopBuildPackageLocation="$(build.artifactStagingDirectory)\WebApp.zip" /p:DeployIisAppPath="Default Web Site"'

    platform: '$(buildPlatform)'

    configuration: '$(buildConfiguration)'

- task: CopyFiles@2

  inputs:

    SourceFolder: 'myAspProject/AzureResourceGroup1'

    Contents: 'azuredeploy\*.json'

    TargetFolder: '$(Build.ArtifactStagingDirectory)'

- task: PublishBuildArtifacts@1

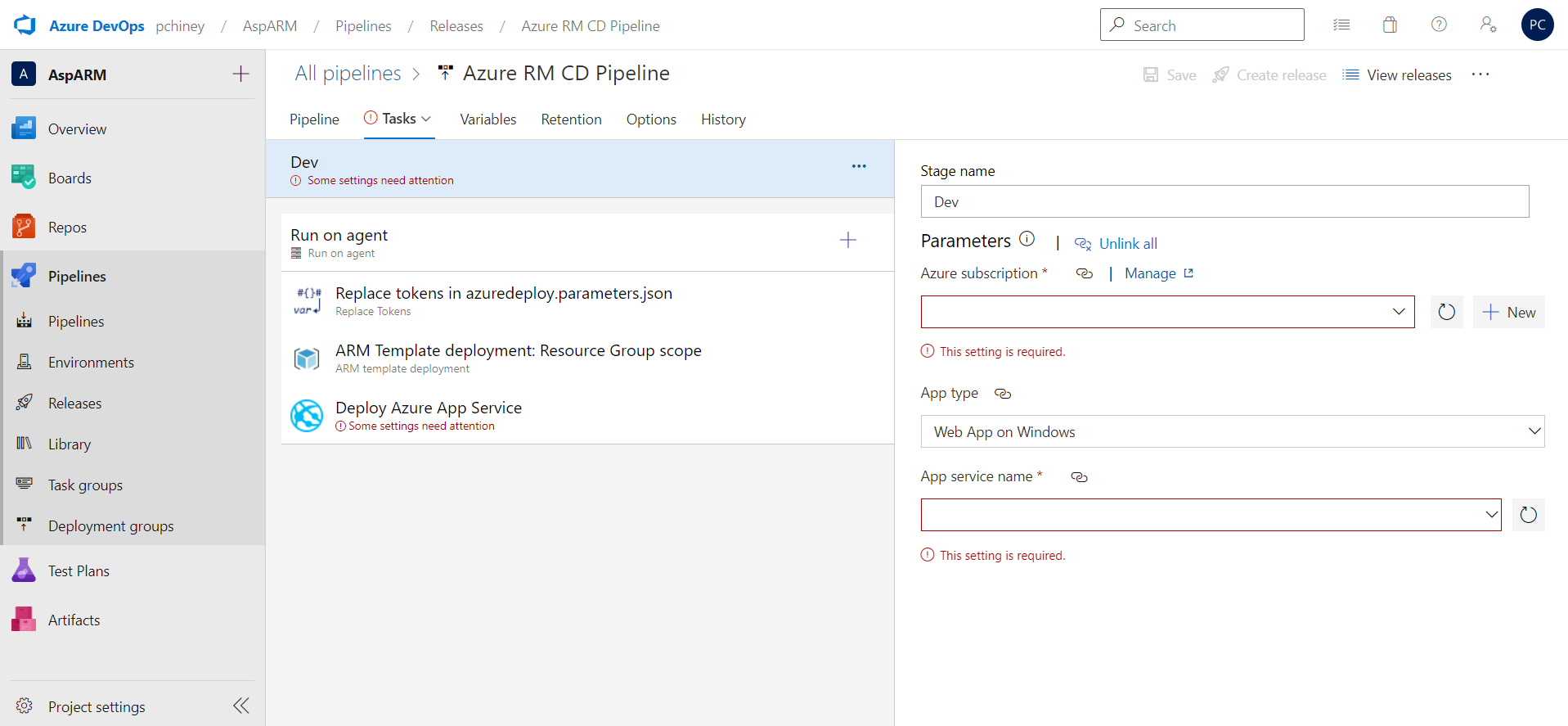
  inputs:

    PathtoPublish: '$(Build.ArtifactStagingDirectory)'

    ArtifactName: 'drop'

    publishLocation: 'Container'

CD:



**MULTI REPO CI:**

# 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

resources:

  repositories:

  - repository: myproject1

    type: git

    name: MultiRepoProject/MultiRepoProject

  - repository: myproject2

    type: git

    name: MultiRepoProject/First repo

trigger:

- main

pool:

  vmImage: ubuntu-latest

steps:

- checkout: myproject1

- checkout: myproject2

- task: CopyFiles@2

  inputs:

    SourceFolder: '$(system.defaultworkingdirectory)'

    Contents: '\*\*'

    TargetFolder: '$(build.artifactstagingdirectory)'

- task: PublishBuildArtifacts@1

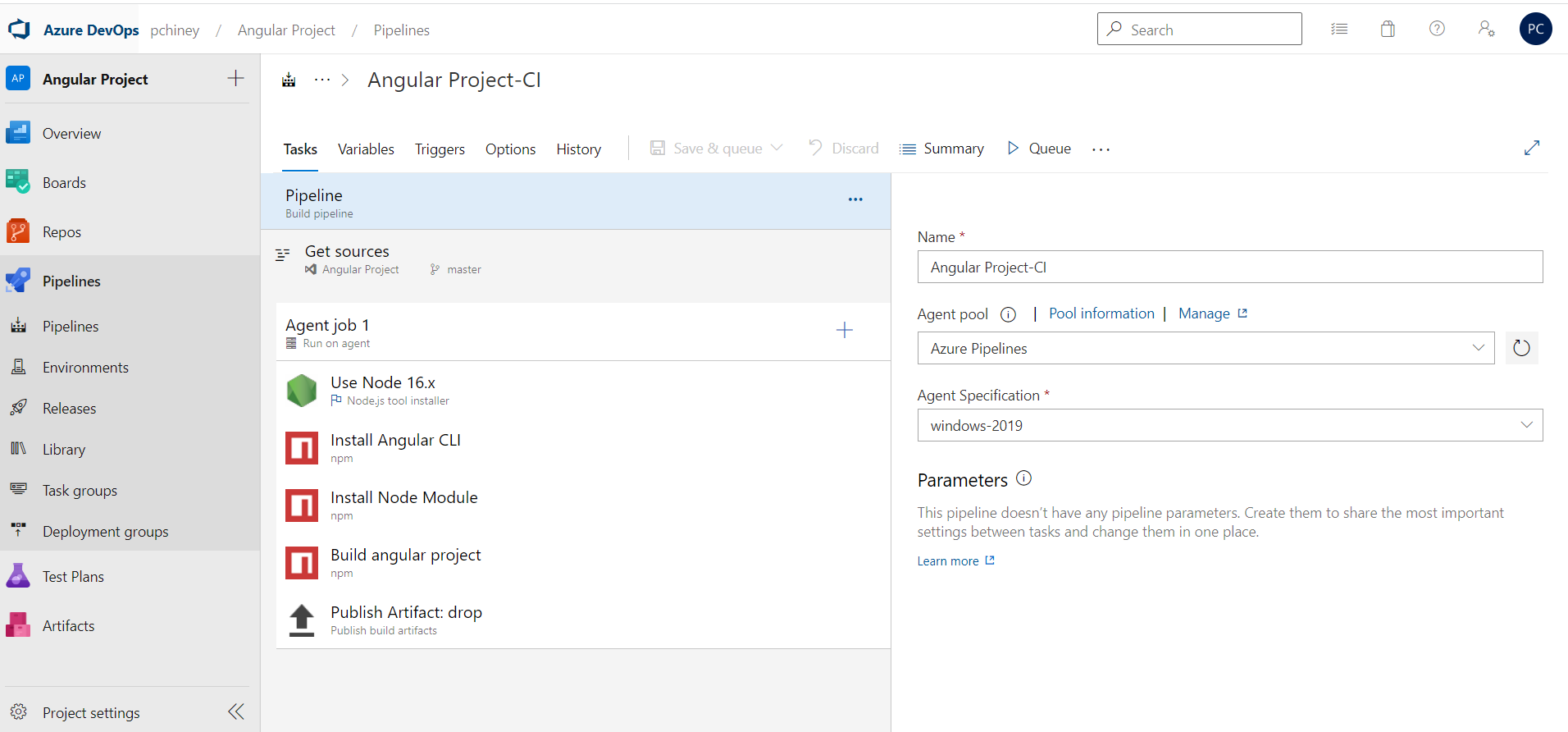
  inputs:

    PathtoPublish: '$(Build.ArtifactStagingDirectory)'

    ArtifactName: 'drop'

    publishLocation: 'Container'

**ANGULAR PROJECT CI:**



pool:

name: Azure Pipelines

steps:

- task: NodeTool@0

displayName: 'Use Node 16.x'

inputs:

versionSpec: 16.x

- task: Npm@1

displayName: 'Install Angular CLI'

inputs:

command: custom

verbose: false

customCommand: 'install @angular/cli@12'

- task: Npm@1

displayName: 'Install Node Module'

inputs:

workingDir: 'angular-tour-of-heroes-pc'

verbose: false

- task: Npm@1

displayName: 'Build angular project'

inputs:

command: custom

workingDir: 'angular-tour-of-heroes-pc'

verbose: false

customCommand: 'run ng build --watch'

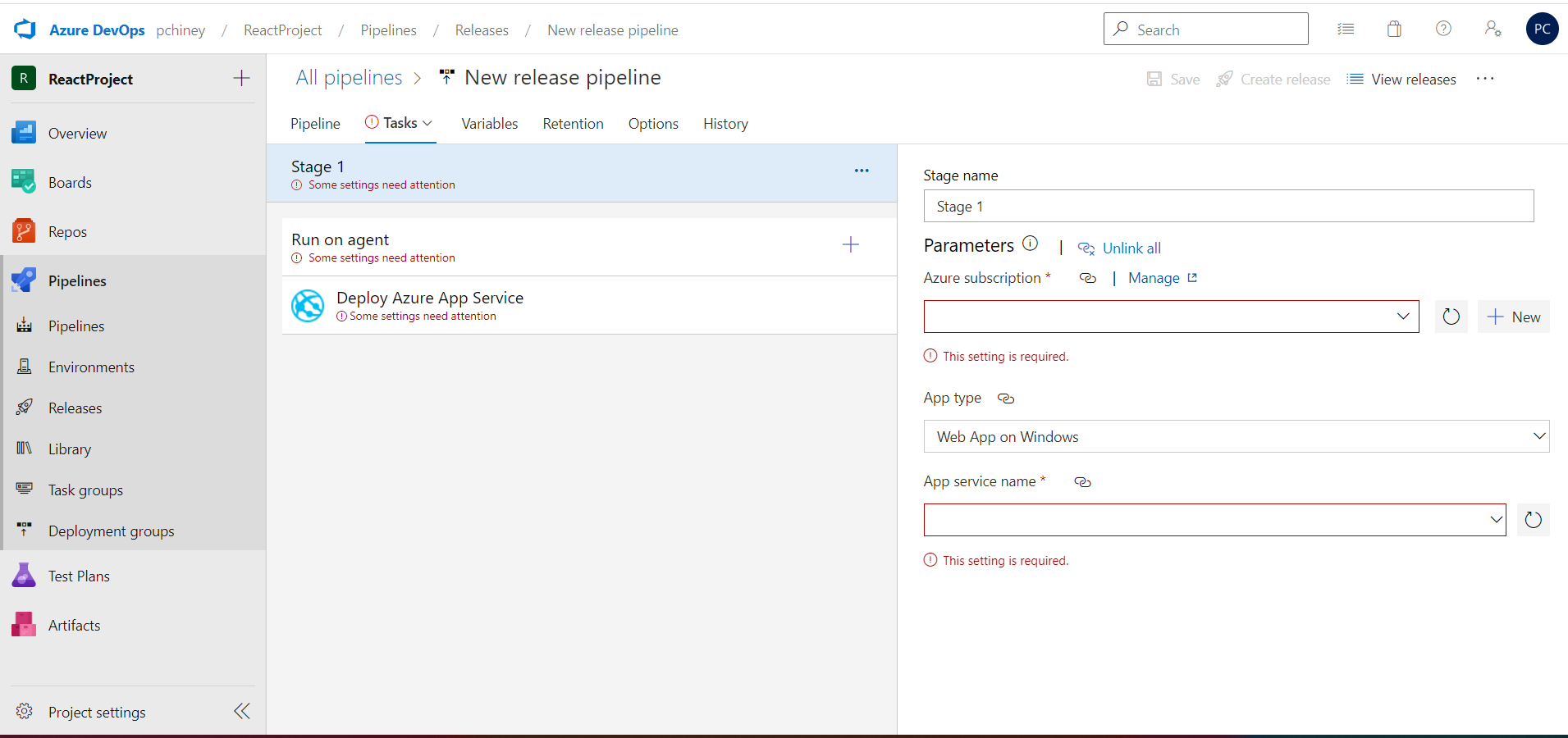
- task: PublishBuildArtifacts@1

displayName: 'Publish Artifact: drop'

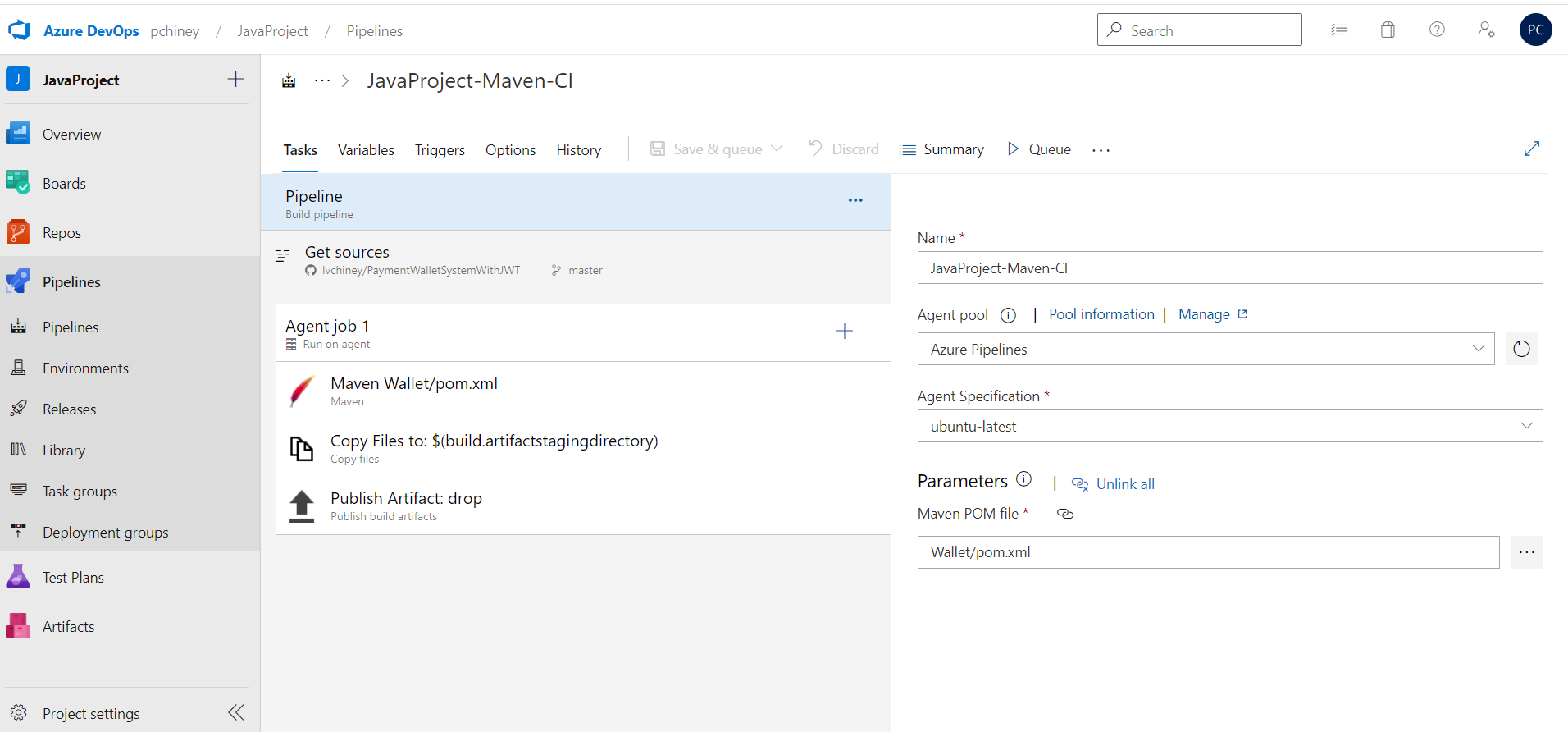
inputs:

PathtoPublish: 'angular-tour-of-heroes-pc/dist'

**ANGULAR PROJECT CD:**



**JAVA PROJECT CI:**



pool:

name: Azure Pipelines

demands: maven

#Your build pipeline references an undefined variable named ‘Parameters.mavenPOMFile’. Create or edit the build pipeline for this YAML file, define the variable on the Variables tab. See https://go.microsoft.com/fwlink/?linkid=865972

steps:

- task: Maven@3

displayName: 'Maven Wallet/pom.xml'

inputs:

mavenPomFile: '$(Parameters.mavenPOMFile)'

- task: CopyFiles@2

displayName: 'Copy Files to: $(build.artifactstagingdirectory)'

inputs:

SourceFolder: '$(system.defaultworkingdirectory)'

Contents: '\*\*/\*.jar'

TargetFolder: '$(build.artifactstagingdirectory)'

condition: succeededOrFailed()

- task: PublishBuildArtifacts@1

displayName: 'Publish Artifact: drop'

inputs:

PathtoPublish: '$(build.artifactstagingdirectory)'

condition: succeededOrFailed()

**JAVA PROJECT CD:**